

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

Université 20 Août 1955-Skikda

جامعة 20 أوت 1955 سكيكدة

Faculté des Sciences

كلية العلوم

Département : Informatique

قسم: الإعلام الآلي



Mémoire présenté en vue de l'obtention

Du diplôme de

Master en Informatique

Option : Génie logiciel avancé et applications (GLAA)

**Développement d'un service web pour l'analyse des
alarmes corrélées.**

Présenté par :

Kharef Sidali

Bekkouche Zakaria

Encadré par :

Bouarroudj Samiya

Année Universitaire 2021/2022

Remerciements

Nous remercions profondément notre encadreur

« BOUARROUDJ SAMIA » pour l'aide précieuse qu'elle nous a
apporté et pour les conseils qu'elle n'a pas manqué de nous
prodiguer afin de pouvoir mener à bien ce travail.

Nous remercions nos familles respectives et particulièrement nos
parents pour leurs soutiens qu'ils nous ont accordés tout au long
de notre chemin.

Dédicace

Avant tout, je remercie « Dieu » le tout puissant pour m' avoir donné la force, le

courage et la chance d' étudier et de suivre le chemin de la science.

C' est avec profonde gratitude et sincères mots, que je dédie ce travail à mes chers

parents ; qui ont sacrifié leur vie pour ma réussite et m' éclaire le chemin par leurs

conseils judicieux.

je dédie aussi ce travail à mon frère adel et ma sœur Amani.

À mes amis Narimen, Hatem, Islem, Wahid, mohamed, aymen

Et à ceux qui ont partagé avec moi tous les moments d' émotion lors de la réalisation

de ce travail.

Sidali

Dédicace

A mes chers parents qui m'ont toujours soutenu et encouragé au cours de

Ces années d'études. Qu'ils trouvent ici le témoignage de ma profonde de

Reconnaissance.

A ma famille, mes proches et à ceux qui me donnent de la vivacité.

Je dédie aussi ce travail à ma sœur Amina & sa petite fille Sophia

A mes cousins Amine et Saci.

A mes amis Hatem, Islam. Wahid

Lakaria

Résumé

L'objectif principal de ce travail est d'améliorer la détection des alarmes corrélées et de quantifier le niveau de corrélation afin de simplifier la gestion des systèmes d'alarme industriels. La méthode a été implémentée dans un service web qui possède trois phases principales. La première phase vise à réduire les alarmes manquées, fausses et intempestives tandis que la deuxième phase consiste à segmenter des événements, dans laquelle nous séparons les séquences d'alarmes. Dans la dernière phase, on utilise les résultats de la deuxième phase pour détecter les corrélations entre les alarmes.

Mot clés

exploration de données ;séquences d'alarmes ;alarmes corrélées ;processus industriel ;Data mining

Abstract

The main goal of this work is to improve the detection of correlated alarms and quantify the correlation level in order to simplify the management of industrial alarm systems. The method has been implemented in a service web that has three main functions. The first and most important function aims to reduce missed, false, and chattering alarms. The second, Event Segmentation in which we separate sequences, hence the name. The third and last function uses the results of the second phase to isolate and detect correlations between alarms.

Keywords

data Mining;alarm sequences;correlated alarms;chemical processes;nuisance alarms

Table des matières

Introduction générale

1. Contexte de la recherche	01
2. Problématique	02
3. Notre contribution	02
4. Description du contenu	03

Chapitre 01

Partie 01 Supervision industrielle

1. Introduction	04
2. Surveillance des procédés industriels	04
2.1. LES SYSTÈMES INDUSTRIELS CONSIDÉRÉS : LES PROCÉDÉS	05
2.2. Procédés industriels : pourquoi les surveiller ?	05
3. Supervision	05
4. La Supervision industrielle	06
5. Le rôle d'un opérateur dans la supervision	07
6. Surveillance	08
7. Le pilotage	09
8. Conclusion	10

Partie 02 Services web

1. Introduction	11
2. Service Web	11
3. Les avantages des services web et les inconvénients	12
3.1. Avantages.	13
3.2. Les inconvénients	13
4. Architecture de référence des services web	13
5. Communication	14
6. Description	17
7. Publication	18
8. Comparaison entre SOAP & REST	20
9. L'intérêt d'un service web	20
10. Caractéristiques des Services web	21
11. Conclusion	22

Chapitre 02 Gestion des alarmes

1. Introduction	23
2. Définition d'une alarme	23
3. Les types d'alarmes DCS.	24
3.1 Alarme de procédé	24
3.2 Alarmes liées à la sécurité	24
3.3 Alarme d'arrêt d'urgence	24
4. Cycle de vie de la gestion des alarmes	25
4.1 Alarm Philosophy (A)	26
4.2 Identification (B)	26
4.3 Rationalization (C)	26
4.4 Detailed Design (D)	27
4.5 Implementation (E)	27
4.6 Operation (F)	27
4.7 Maintenance (G)	27
4.8 Monitoring and Assessment (H)	27
4.9 Management of Change (I)	28
4.10 Audit (J)	28
5. Gestion des alarmes	28
6. La chaudière industrielle (cas d'étude)	29
6.1 Définition d'une chaudière	29
6.2 Principaux éléments d'une chaudière	29
6.2.1 Les équipements	30
6.2.2 Les instruments	32
7. Description fonctionnelle	34
8. Valeur de conception de référence	34
9. Description de l'architecture des systèmes	35
10. Conclusion	35

Chapitre 03 Conception

1. Introduction	36
2. Objectif du travail	36

3. Travaux existants	36
4. Architecture du système.	37
4.1. Interaction Client- Serveur	38
4.2. Prétraitement	38
4.3. Attribution des valeurs temporelles et segmentation des séquences	40
4.4. Apprentissage	42
5. Etude de cas	43
5.1. Prétraitement	43
5.2. Segmentation des séquences	48
5.3. Apprentissage	49
6. Validation	50
6.1. Précision	50
6.2. Rappel	51
6.3. Calcul de la précision et du rappel	51
6.4. Discussion	53
7. Conclusion	53

Chapitre 04 Implémentation

1. Introduction	54
2. Les outils utilisés dans le développement	54
2.1. Visual Studio Code	54
2.2. Nodejs	55
2.3. React	55
2.4. Express	55
2.5. ESLint	55
2.6. Anaconda	56
2.7. Python	56
2.8. Les bibliothèques utilisées	56
2.8.1. Numpy	56
2.8.2. Pandas	56
2.8.3. Matplotlib	57
2.8.4. Json	57
2.8.5. Flask	57
2.8.6. Seaborn	57
3. Les étapes de création de service web	57

3.1.Backend (Flask)	57
3.2.Front-End (ReactJS & ElectronJS)	61
3.3.Quelques interfaces de service web	64
4. Conclusion	66
Conclusion générale et perspectives	67

Liste des figures

Figure 1.1: Représentation du modèle fonctionnel de la supervision	06
Figure 1.2: La boucle de supervision	08
Figure 1.3: Relation entre surveillance et supervision	09
Figure 1.4: Description des services web	12
Figure 1.5: Architecture des services web	14
Figure 1.6: Concepts du protocole SOA	16
Figure 1.7: Structure d'un message SOAP	16
Figure 1.8: Structure d'un document WSDL	18
Figure 1.9: Structure de données au sein de l'UDDI	19
Figure 1.10: Les trois pages de l'annuaire UDDI	20
Figure 2.1: Alarm Management Life cycle	25
Figure 2.2: Diagramme P&ID de la chaudière	30
Figure 2.3: Economiseurs de la chaudière	31
Figure 2.4: Présentation du Sur chauffeur	31
Figure 2.5: Les instruments de la chaudière	33
Figure 3.1: Architecture du système.	37
Figure 3.2: architecture des services web	38
Figure 3.3: Exemple des méthodes de segmentation	40
Figure 3.4: Historique avant sélection de la chaudière.	43
Figure 3.5: Historique après sélection de la chaudière.	44
Figure 3.6: Avant suppression des alarmes parasites.	45
Figure 3.7: Après suppression des alarmes parasites.	46
Figure 3.8: Alarmes redondantes	46
Figure 3.9: Codification des Tags	47
Figure 3.10: Etat de l'alarme	47
Figure 3.11: Attribution valeur temporelle	48
Figure 3.12: montre la sortie de l'algorithme 2 (liste des transactions de l'alarme 47).	48

Figure 3.13: montre la sortie de l'algorithme 3 (liste des transactions de l'alarme 47,T=3).	49
Figure 3.14: Résultat de l'apprentissage	49
Figure 3.15: Résultats de précision et de rappel du modèle d'apprentissage.	52
Figure 4.1: Les outils utilisés dans le développement	54
Figure 4.2: Création d'un nouvel environnement Conda nommé AlarmService	58
Figure 4.3: activation de l'environnement AlarmService	58
Figure 4.4: Classes et fichiers back-end	58
Figure 4.5: Code source de la classe « app.py »	59
Figure 4.6: Démarrage du serveur	60
Figure 4.7: PostMan : Résultat de test back-end preprocessing	60
Figure 4.8: Front-end files	61
Figure 4.9: App.js code source	62
Figure 4.10: Preprocessing window Code source	63
Figure 4.11: « package.js » configuring backend/frontend	64
Figure 4.12: La fenêtre principale de service.	64
Figure 4.13: La fenêtre du preprocessing.	65
Figure 4.14: La fenêtre paramètre	65

Liste des Tableaux

Tableau 1.1 : Exemple de la méthode REST	12
Tableau 3.1 : Exemples des alarmes systèmes.	44
Tableau 3.2 : Exemples des modes opératoires	45
Tableau 3.3 : Explication des symboles des formules Precision et Recall	51
Tableau 3.4 : Résultats de notre algorithme	51

Liste des abréviations

IC	Indicator Controller (stay for F=Flow, L=Level; P=Pressure; T=Temperature; H=Hand; etc..)
PV	Process Variable (controller signal input)
SP	Set Point of controller
OUT	Controller signal Output
MAN/AUTO	Manual-Automatic Station of controller
OP	Operation Mode of controller (MAN (manual)-AUTO (automatic)-TRK (tracking))
OS	Operating Station of DCS (or also named HMI)
ACTION	Regulating action of controller DIR (Direct=PV increase Output increase) REV (Reverse=PV increase Output decrease)
ALARMS	Alarm Threshold of controller HIGH (value higher than PV) or LOW (value lower than PV)
FY	Function Block (all type of formula or equation)
P; I; D	Regulation internal algorithms of controller (P=Proportional; I=Integral; D=Derivative) a controller can be only P; or P+I; or P+I+D
NR	Normal
OOP	Sortie Ouverte.
IOP	Entrée Ouverte
HH	Alarme seuil Très Haut sur la mesure.

LL	Alarme seuil Très Bas sur la mesure.
HI	Alarme seuil Haut sur la mesure.
LO	Alarme seuil Bas sur la mesure.
DV	Alarme sur l'écart.
VEL	Alarme de Vitesse sur la mesure.
MHI	Alarme limite Haute sur la sortie.
MLO	Alarme limite Basse sur la sortie.
ANS	Alarme de discordance.
PERR	Erreur de Parité sur les recopies.
ALM	Annonciateur en alarme.
AOF	Alarmes Masquées.
MAN	Manuelle
AUT	Automatique
IMAN	Initialisation Manuel
TRK	Sortie Suiveuse.
CAS	Cascade.
CAL	Calibration.
PV	Mesure (valeur venant du procédé).

Introduction générale

1. Contexte de la recherche

Les installations industrielles modernes sont constituées de nombreux capteurs, actionneurs et boucles de régulation hautement interconnectés. En raison de la complexité du système, une seule défaillance produite dans un seul composant pourra se propager dans tous les dispositifs de l'usine provoquant ainsi des perturbations opérationnelles, voire des conséquences catastrophiques. Cependant les systèmes d'alarme jouent un rôle fondamental dans le maintien de la sécurité des productions industrielles. Pour cela de nombreuses méthodes sont développées en vue de la détection et le diagnostic des défaillances. Ces méthodes se différencient par rapport au type de connaissances à priori sur le processus qu'elles nécessitent. On peut ainsi les classer comme méthodes à base de modèles et méthodes à base de données historiques.

La gestion des alarmes nécessite des méthodes d'apprentissage à base de données historiques. Elle fait référence à la conception, à la mise en œuvre, à l'exploitation et à la maintenance efficaces des alarmes des usines de fabrication industrielle. La gestion des alarmes est nécessaire dans une usine de production contrôlé par un opérateur humain utilisant un système de contrôle, tel qu'un DCS (Distributed control system) et un contrôleur logique programmable (PLC). Un DCS est un système de contrôle d'un procédé industriel doté d'une interface homme-machine pour la supervision et d'un réseau de communication numérique. La gestion des alarmes vise essentiellement à prévenir, ou du moins à minimiser, les pertes physiques et économiques par l'intervention de l'opérateur en réponse à la condition qui a fait l'objet de l'alarme.

Les services web permettent de changer la nature du web, au web orienté services utilisés pour accomplir une tâche donnée. L'évolution d'internet et la compétitivité entre les entreprises ont été le facteur de l'explosion des services disponible sur le web et ils ont permis de développer des nouveaux paradigmes d'interaction entre applications. Notamment, l'architecture orientée service (Service-Oriented Architecture, ou SOA) a été mise en avant afin de permettre des interactions entre applications assez distantes.

2. Problématique

Les systèmes d'alarme sont des atouts essentiels des installations industrielles modernes pour aider les opérateurs à gérer les perturbations et les situations dangereuses. Cependant en raison de la forte corrélation entre de nombreuses variables de processus, de nombreuses alarmes séquentielles se produisent, ce qui peut conduire à une inondation d'alarmes. Dans cette situation, il est difficile pour les opérateurs de traiter les alarmes en temps voulu et efficacement, et peut également les rendre moins attentifs aux alarmes significatives.

3. Contribution

Dans une situation d'inondation d'alarmes, un grand nombre d'alarmes se produisent en peu de temps et les opérateurs ne peuvent pas traiter les informations d'alarme de manière efficace. Par conséquent, la situation anormale liée à l'alarme est négligée, ce qui a un impact négatif sur la sécurité ou l'efficacité des dispositifs industriels. Dans ce cas, grâce à l'analyse de corrélation des alarmes, les alarmes racines peuvent être trouvées et traitées efficacement, ce qui est utile pour supprimer l'inondation des alarmes. Parallèlement, les alarmes corrélées peuvent être regroupées et présentées aux opérateurs comme des symptômes permettant de détecter des anomalies.

Notre travail consiste à créer un service web pour l'aide à la décision. Il permet, à partir des journaux d'alarmes issus d'un DCS, de détecter les alarmes corrélées et quantifier leurs niveaux de corrélation au sein du complexe RA1K Sonatrach Skikda. Le service développé peut s'exécuter à distance et donc permet à l'opérateur de prendre des décisions décentralisées. L'aspect distribué permet aussi à d'autres experts de contribuer à résoudre les problèmes sérieux afin de minimiser les risques.

Le service Web permet aux applications client-serveur de communiquer sur un réseau. Le fonctionnement côté client consiste à:

- Prendre connaissance des interfaces que le service Web propose;
- Construire la requête;
- Envoyer la requête;
- Récupérer et interpréter les données retournées (JSON, XML ou TEXT) ;
- Traiter les données (calculer, afficher) ;

Tandis que le fonctionnement côté serveur permet de :

- Définir les interfaces publiques;
- Récupérer les requêtes du client;
- Traduire la requête et effectuer le traitement;
- Envoyer la réponse normalisée dans un format standard (JSON, XML ou TEXT) ;

4. Description du contenu

Notre mémoire se compose de quatre chapitres. Dans le premier chapitre, nous décrivons des concepts généraux concernant les services web. On commencera par décrire ses les principales technologies de développement ainsi que ses caractéristiques. Puis on va présenter l'intérêt, les avantages et les inconvénients des services web. Dans le deuxième chapitre nous allons présenter et décrire les alarmes en général et élaborer la gestion de ces derniers précisément. Dans le troisième chapitre Nous allons présenter un modèle qui décrira la réalisation de notre système ainsi que son fonctionnement. Et dans le quatrième et dernier chapitre nous présenterons les outils et langages de programmation utilisés pour arriver à notre but, nous allons également offrir des figures qui montrent les différentes fenêtres de notre système.

Supervision

1. Introduction

La 4ème révolution industrielle ou l'industrie 4.0 est le sort de l'ensemble des évolutions qu'a connu le monde de l'industrie. Car la demande du consommateur est toujours en augmentation avec l'exigence d'une qualité meilleure à bas prix. C'est là qu'une révolution technologique apparaîtra et tend à modifier fondamentalement notre façon de vivre, de travailler et d'établir des relations entre nous. Cette industrie comprend des groupes technologiques tels que les robots autonomes, l'internet des objets (IoT), des simulations (Jumeau numérique), l'intégration de systèmes horizontaux et verticaux, l'infonuagique (big data), la cybersécurité, et la réalité augmentée ou virtuelle qui apportent des changements de grande envergure, aux entreprises, non seulement par l'intégration et l'interconnectivité des personnes, des machines, et des organisations, mais touche également aux modèles d'affaires et de création des valeurs, les processus et activités et le management organisationnel des entreprises [6].

Aujourd'hui nous sommes à l'ère de l'industrie du futur, ou la digitalisation des processus industriels a pour but de transformer les industries pour une meilleure gestion en temps-réel de la production, de la maintenance, de l'énergie ou encore de tous les flux logistiques. Notamment grâce à la supervision qui est une technique industrielle de suivi et de pilotage informatique de procédés de fabrication automatisés, elle concerne l'acquisition de données (mesures, alarmes, retour d'état de fonctionnement) et des paramètres de commande des processus pour les suivre en temps réel que ce soit en fonctionnement normal ou en présence de défaillances, et elles sont généralement confiés à des automates programmables pour maintenir le procédé dans un mode de fonctionnement optimal. [1]

2. Surveillance des procédés industriels

Nous nous intéressons ici à la surveillance et au diagnostic de machines, de structure ou de procédés, que nous désignerons le plus souvent par le terme de systèmes et qui sont essentiellement dynamiques. Nous indiquons maintenant ce que nous entendons par système dynamique et les raisons qui conduisent à le surveiller.

2.1. Les systèmes industriels considérés : les procédés

Un procédé industriel est un procédé de nature mécanique ou chimique destiné à produire des objets ou à synthétiser des produits chimiques en grande quantité et dans des conditions techniquement et économiquement acceptables. Ils sont notamment essentiels aux industries dites lourdes.

Le Génie des procédés consiste à étudier, concevoir et mettre en œuvre de façon optimale les procédés, systèmes industriels dynamiques (complexes) de transformation de la matière et de l'énergie. Cette discipline a ainsi apporté des concepts novateurs dans des domaines d'application très variés.

2.2. Procédés industriels : pourquoi les surveiller ?

L'étude bibliographique menée nous a permis d'identifier les besoins réels de la surveillance et de la supervision. Nous nous proposons maintenant d'exposer ces besoins :

- Nécessité de disposer de toutes les fonctions de surveillance et de supervision
- Nécessité de proposer plusieurs traitements de défaillances (flexibilité des traitements) ;
- Nécessité de disposer d'une bibliothèque de l'ensemble des traitements de surveillance ;
- Nécessité de s'appuyer sur une technique de synthèse des stratégies de surveillance-supervision ;
- Nécessité de disposer d'une technique de détection de défaillances qui ne soit pas basée sur le recensement exhaustif des défaillances de la partie opérative. L'identification de toutes les situations de défaillance qui peuvent se présenter au niveau d'un procédé complexe peut devenir très difficile, voire impossible. Une technique générique de caractérisation des symptômes de défaillances est donc requise ;

Nécessité d'intégrer dès la phase de conception les modes de marches et d'arrêts. Il va de soi qu'une défaillance sera certainement traitée différemment selon que le procédé se trouve en fonctionnement normal ou dégradé. [5]

3. Supervision

La supervision d'un procédé est l'interface entre la gestion hors ligne et les systèmes de commande. Elle consiste en la conduite d'installations industrielles tant en fonctionnement normal qu'en présence de défaillances. Elle permet de suivre en temps réel des informations disponibles et ainsi de réagir face aux variations observées pour maintenir le procédé dans un

mode de fonctionnement optimal. La supervision englobe donc les tâches de pilotage et de surveillance en temps réel d'un procédé complexe, selon un certain degré d'abstraction [1].

Les principaux objectifs des systèmes de supervision :

- Concentrer les données, déporter ou centraliser le pilotage de procédé ;
- Une vision en temps-réel de la production ainsi que du procédé permettant aux opérateurs de réagir et de décider rapidement, apporter les premiers outils d'analyse nécessaires aux contrôles des équipements concerné (historique, courbes, alarme...) ;

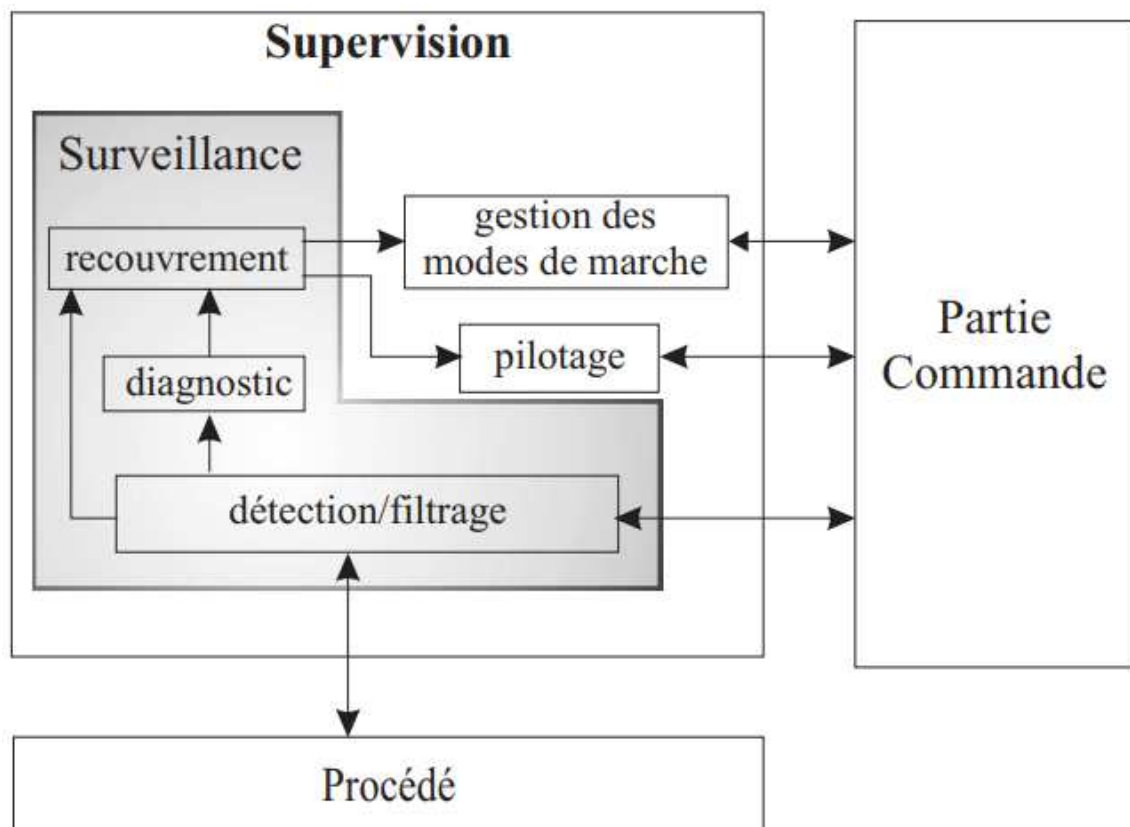


Figure 1.1 : Représentation du modèle fonctionnel de la supervision [3]

4. La Supervision industrielle

La supervision est un excellent moyen pour toutes les industries ayant des besoins de pilotage et de visualisation de l'état de leur équipement. Ces outils ont pour but la gestion en temps-réel de la production avec une représentation graphique et une prise en charge de fonction avancées du procédé pour un mode de fonctionnement optimal. Les systèmes de supervision

permettent d'avoir une vue synthétique des équipements ou ensembles d'équipement afin de visualiser leur état physiques ou fonctionnels, situés dans des salles de commande loin des équipements, ça permet d'ailleurs d'offrir la possibilité de déporter et de centraliser la vision et le pilotage des organes physiques (capteurs, actionneurs) qui sont parfois très éloignés.

Les principaux objectifs des systèmes de supervision :

- Concentrer les données, déporter ou centraliser le pilotage de procédé ;
- Une vision en temps-réel de la production ainsi que du procédé permettant aux opérateurs de réagir et de décider rapidement, apporter les premiers outils d'analyse nécessaires aux contrôles des équipements concerné (historique, courbes, alarme...);

5. Le rôle d'un opérateur dans la supervision

Malgré la montée en puissance de l'automatisation dans l'industrie, l'opérateur se positionne toujours comme le maillon intelligent de la boucle de surveillance : il est chargé d'analyser la situation et de prendre les décisions appropriées. De plus, après une panne ou une dérive, la réponse est généralement manuelle ou Semi-automatique. La mise en place d'une interface ergonomique facilite ses tâches d'analyse, donnant différentes vues du processus sous forme de vue d'ensemble. Ceux-ci reproduisent l'installation et affichent les grandeurs mesurées en temps réel. [4]

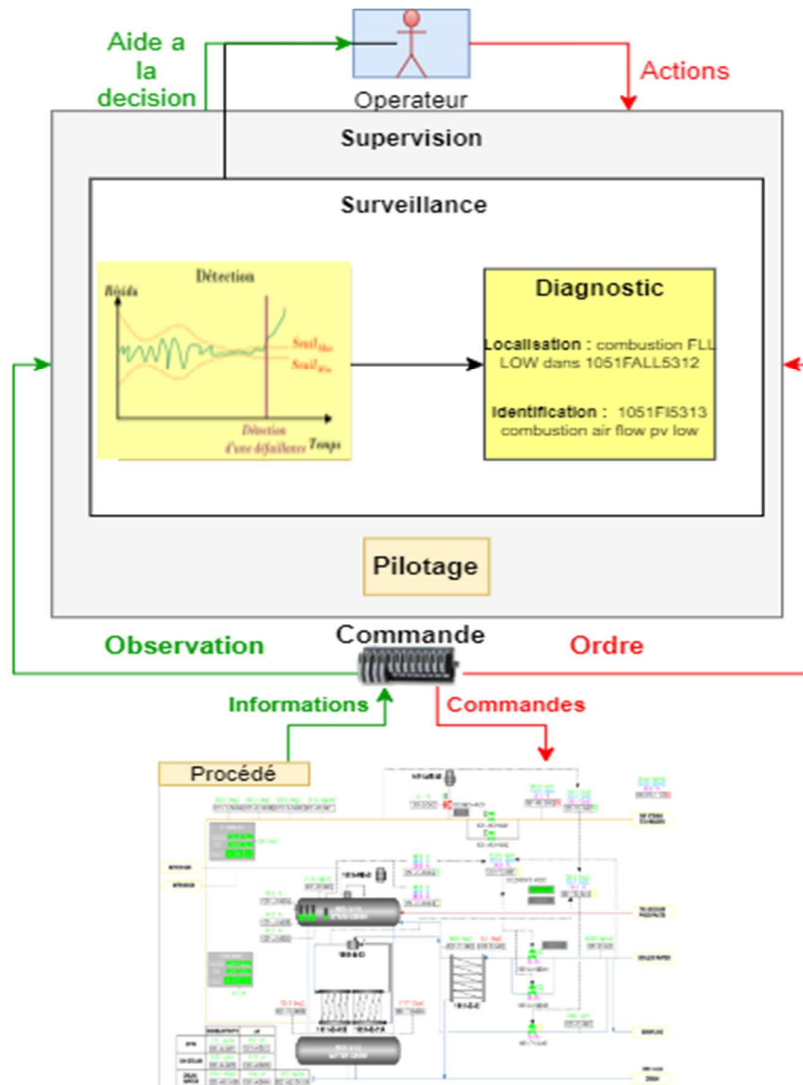


Figure 1.2 :La boucle de supervision

6. Surveillance

La surveillance est chargée de recueillir en permanence tous les signaux en provenance du procédé et de la commande, de suivre en temps réel les évolutions du système commandé, de faire toutes les inférences nécessaires pour dresser des historiques de fonctionnement, et, le cas échéant, pour mettre en œuvre un processus de traitement de défaillance et générer des alarmes à partir des informations délivrées par des capteurs, des seuils sont définis sur des variables clés par des experts du procédé selon des critères de sécurité concernant les hommes, l'installation et son environnement. La génération d'alarmes apporte une aide considérable aux opérateurs humains de supervision dans leur tâche de surveillance afin qu'ils puissent analyser la situation et prendre la meilleure décision adaptée (procédure d'arrêt d'urgence, mode dégradé, action corrective). [3]

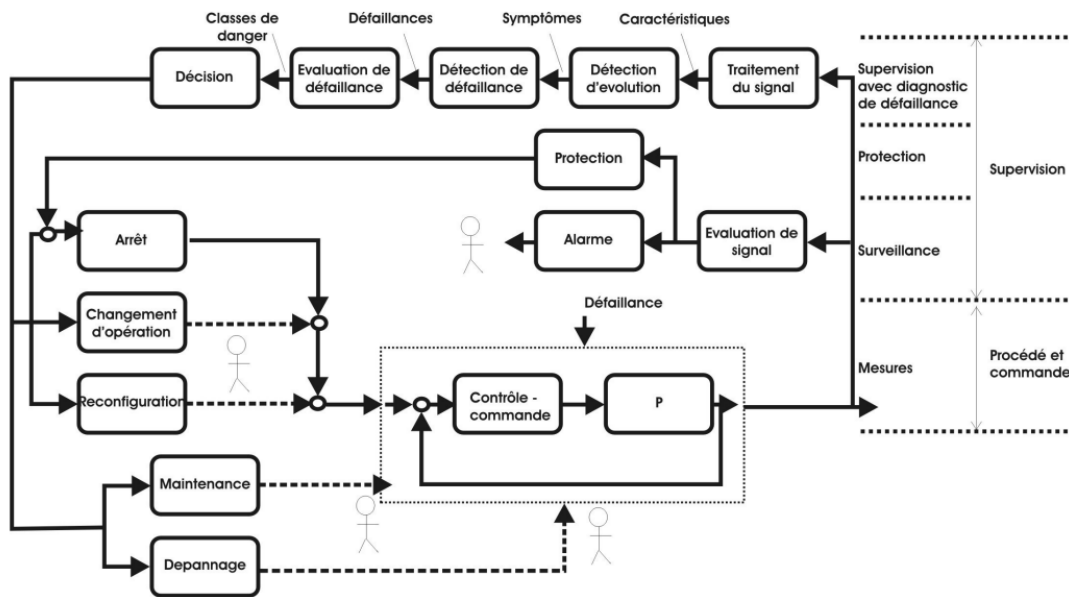


Figure 1.3 : Relation entre surveillance et supervision [2]

7. Le pilotage

Le pilotage en temps réel inclut l'exécution de la séquence d'opérations pour chaque recette et assure la gestion en temps réel des ressources nécessaires pour effectuer cette exécution. Pour cela, il met en œuvre en temps réel les ordres de fabrication (consignes) de la fonction d'ordonnancement en convertissant ces informations en tâches de commande et en les distribuant aux automates et contrôleurs locaux.

Par conséquent, cette fonction se trouve au-dessus du système de contrôle et est responsable de :

- Gérer (démarrer, arrêter, mettre en pause...) des règles de pilotage prédéfinies (programmes automates Paramètres de régulation, etc.) ;
- Gérer les transitoires (démarrages, changements de point de fonctionnement, arrêts) qui correspondent aux changements d'état du système. ;

Cette caractéristique induit souvent un désir d'autonomie de l'opérateur (source de répétabilité) et de réactivité par rapport au mode opératoire. C'est une fonction très automatisée. Le pilotage vise principalement les normes de qualité de la productivité et la tenue des délais.

L'objectif est d'augmenter la maîtrise des coûts de production et de stockage, et de raccourcir les temps de réponse en cas de perturbation. Pour les deux premiers objectifs, l'amélioration des modes opératoires et une plus grande fiabilité des échanges d'informations

contribue fortement à l'optimisation des coûts de production et de stockage. En ce qui concerne l'amélioration du temps de réaction en cas de perturbation, deux solutions doivent être envisagées : prévoir des actions correctives de planification et de programmation hors ligne pour répondre aux problèmes liés à l'évolution du marché (indisponibilité temporaire de matières premières, modification des spécifications client, etc.), en ajoutant toutes compétences requises dans les couches du niveau en temps réel pour mieux résoudre les problèmes qui peuvent survenir pendant la phase de transformation des produits (dysfonctionnements des machines, modification des commandes, etc.).

8. Conclusion

Dans cette partie nous avons présenté la surveillance et supervision industrielles ainsi que leurs différentes importances. Dans la partie suivante nous présenterons les services web ainsi que leurs différentes caractéristiques

Service Web

1. Introduction

De nos jours, les entreprises et les systèmes d'information expriment un grand besoin pour échanger des informations et des services. Ceci nécessite des langages communs de communication. Les efforts de standardisation de ces langages ont donné lieu à un nouveau domaine de recherche connu sous le nom de « protocoles B2B ». Une technologie émergente dans ce domaine a permis de tracer quelques pistes intéressantes pour la communication entre entreprises. Cette technologie est celle des services Web. Les services Web sont un paradigme naissant qui vise à la transposition des architectures par composant dans le cadre du Web. Un service Web est un composant logiciel qui offre des services à travers une interface standardisé. La particularité des services Web est l'utilisation de la technologie Internet comme infrastructure pour la communication entre eux. Cette dernière représente une solution émergente et prometteuse pour le développement, le déploiement et l'intégration des applications capables de répondre aux besoins des utilisateurs.

2. Service Web

Le Service Web est un ensemble de normes et de méthodes de programmation pour partager des données entre différentes applications logicielles, en distribuant des services sur Internet qui prend en charge l'interopérabilité du système [7] [8]. Les services Web incluent essentiellement tout logiciel, application ou technologie cloud qui fournit un protocole Web standardisé (HTTP ou HTTPS) pour interagir, communiquer et échanger des messages de données généralement XML (Extensible Markup Language). En d'autres termes, les services Web sont des systèmes d'échange de données centrés sur XML qui utilisent Internet pour la communication et l'interfaçage A2A (application à application).

Une caractéristique clé des services Web est que les applications peuvent être écrites dans différentes langues et être toujours capables de communiquer en échangeant des données entre elles via le service Web entre le client et le serveur. Le client envoie une demande via XML pour appeler le service Web, et le service répond avec une réponse XML. Les services Web

sont également souvent associés à la SOA (Service-Oriented Architecture). Un service Web comprend ces fonctions essentielles :

- Disponible sur les réseaux Internet ou Intranet ;
- Système de messagerie XML standardisé ;
- Indépendant d'un seul système d'exploitation ou langage de programmation
- Auto-description via le langage XML standard ;
- Découvrable grâce à une méthode de localisation simple ;

Un service Web prend en charge la communication entre de nombreuses applications avec HTML, XML, WSDL, SOAP et d'autres normes ouvertes. XML balise les données, SOAP transfère le message et WSDL décrit l'accessibilité du service.

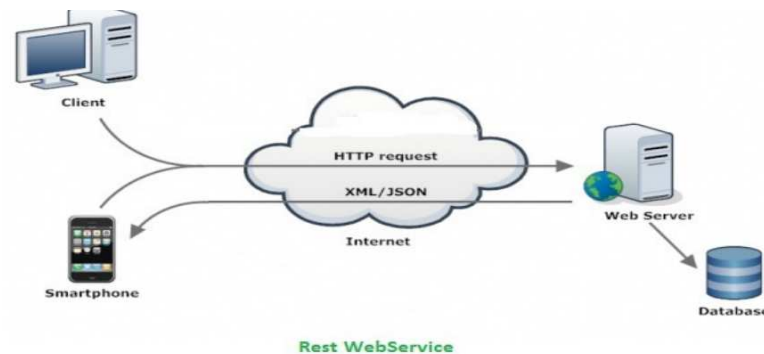


Figure 1.4 : Description des services web [13]

3. Les avantages des services web et les inconvénients

Aujourd'hui, les services web jouent un rôle majeur dans de nombreux secteurs car ils offrent beaucoup d'avantages mais comme chaque technologie ils présentent aussi des inconvénients, nous allons citer quelques avantages et inconvénients :

3.1. Avantages

- Les services web réduisent le temps de mise en marché des services offerts par les diverses entreprises ;
- Les services web permettent à des programmes écrits en des langages différents et sur des plates-formes différentes de communiquer entre eux par le biais de certaines normes (Wolter, 2001). En d'autres termes, les services web permettent une meilleure interopérabilité entre les logiciels ;

- Les services web utilisent des normes et protocoles ouverts ;
- Grâce au protocole HTTP, les services web peuvent fonctionner malgré le pare-feu sans pour autant nécessiter des changements sur les critères de filtrage ;
- Les protocoles et les formats de données sont offerts, le plus possible, en format texte pour que la compréhension du fonctionnement des échanges soit plus intuitive ;
- Grâce aux services web, les coûts sont réduits par l'automatisation interne et externe des processus commerciaux [10] ;

3.2. Les inconvénients

- Les normes de services Web dans certains domaines sont actuellement récentes ;
- Les services Web souffrent de performances faibles comparée à d'autres approches de l'informatique répartie telles que le RMI, CORBA, ou DCOM ;
- Par l'utilisation du protocole HTTP, les services Web peuvent contourner les mesures de sécurité mises en place au travers des pare-feu ;
- Les services web ne sont pas sécurisés totalement. [11] ;

4. Architecture de référence des services web

L'architecture des services web repose sur un mécanisme de transmission de requêtes de service entre un client et un serveur, tous deux connectés à un réseau. Dans cette architecture, le client peut être un navigateur Web (les demandes de service proviennent directement d'une intervention humaine) ou une application (les demandes de service sont alors automatisées). Le bus de requête est basé sur TCP/IP et http (mais aussi sur SMTP ou POP), permettant de l'utiliser sur Internet pour intégrer des applications dans le réseau interne d'une entreprise ou pour "publier" des applications préexistantes sur le réseau à des fins d'utilisation par les entreprises partenaires. HTTP ne transmettant que du texte, tous les échanges entre services web (requêtes et résultats de requêtes) bouclent également au format texte, sous forme de documents encodés en XML (requêtes, leurs résultats et les éventuelles erreurs produites par l'appel). L'architecture de référence est constituée des éléments suivants :

- **Le fournisseur de service** : Le fournisseur de service crée le service web, publie son interface et décrit le service en termes de mise en œuvre ainsi que des informations d'accès au service en WSDL dans un registre de services web UDDI et implémente le service dans un serveur d'applications tel qu'apache ;

- **Le client** : C'est n'importe quel consommateur du service Web. Le demandeur utilise un service Web existant en ouvrant une connexion réseau et en envoyant une demande en XML (REST, XML-RPC, SOAP) ;
- **L'annuaire des services** : Le registre de service est un annuaire de services. Le registre fournit un endroit central où les programmeurs peuvent publier de nouveaux services ou en trouver ;

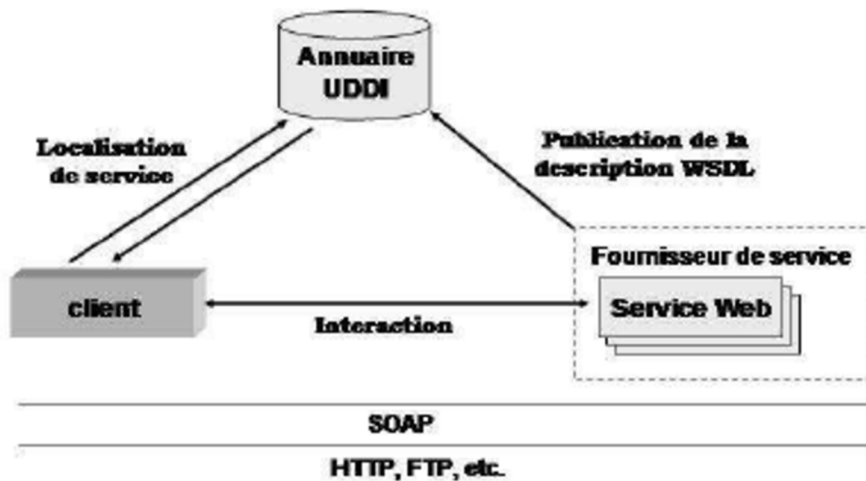


Figure 1.5 : Architecture des services web [12]

5. Communication

Dans les architectures des services Web il existe plusieurs architectures de services Web qui peuvent être utilisés : XML, REST, SOAP, WSDL, UDDI et HTTP.

5.1. XML

XML (Extensible Markup Language, ou Langage Extensible de Balisage) est le langage destiné à succéder à HTML. Comme HTML (Hypertext Markup Language) c'est un langage de balisage (markup) : il représente de l'information encadrée par des balises. XML a avec HTML un ancêtre commun : le SGML (Standard Generalized Markup Language). L'une de ses caractéristiques était la séparation du formatage et du contenu. En effet, le format décrit indépendamment du contenu du document permettait d'obtenir un rendu identique pour une feuille de n'importe quel format. Ce principe est appliqué dans le sens où les données et le schéma du document sont séparés, le schéma représentant la structure et les types de données, incluant la sémantique (importante pour que le document puisse interagir avec différents langages de programmation et systèmes). Les caractéristiques essentielles de XML sont :

- Séparation stricte entre contenu et présentation ;
- Simplicité, universalité et extensibilité ;
- Format texte avec gestion des caractères spéciaux ;
- Structuration forte ;
- Modèles de documents (DTD et Schémas XML) ;
- Format libre ;

5.2. SOAP

Le service Web avec l'architecture SOAP est un protocole de service Web basé sur XML pour échanger des données et des documents via HTTP ou SMTP (Simple Mail Transfer Protocol). Il est maintenu par le groupe W3C. Il permet à des processus indépendants fonctionnant sur des systèmes disparates de communiquer à l'aide de XML. La Figure 1.6 présente, de façon hiérarchique, les principaux concepts ayant trait à ce protocole. D'abord, le bloc « transport » représente le protocole applicatif de communication. Tout protocole non-propretaire pouvant transporter un document XML peut être employé à ce niveau. Dans la quasi-totalité des cas, les protocoles HTTP et HTTPS sont retenus en raison de leur simplicité et de leur support déjà présent dans les entreprises. Toutefois, il arrive que des protocoles asynchrones ou propriétaires, tel JMS, soient employés. Le bloc XML indique que le message SOAP est encodé à l'aide du langage XML. Ceci a une grande importance, car le langage XML utilise les schémas XSD qui permettent de définir des métadonnées pour décrire les types d'objets échangés dans les messages. Le bloc « messagerie » représente, du point de vue des affaires, le document échangé lors de la communication. Finalement, les blocs « fiabilité », « sécurité » et « transaction » représentent des aspects pouvant être implémentés à l'aide des en-têtes des messages SOAP. Ces en-têtes constituent un mécanisme d'extension du protocole et sont représentés à la Figure 1.7 qui montre la structure d'un message SOAP. Le nombre d'en-têtes n'est pas fixe et ceux-ci peuvent être de n'importe quelle nature, pourvu qu'ils soient accompagnés de métadonnées. Si le récepteur ne reconnaît pas un en-tête, il n'est pas forcé de le traiter.

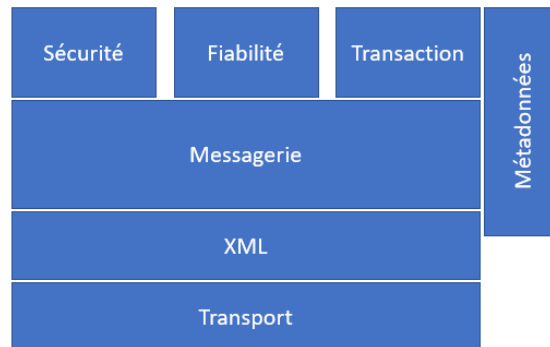


Figure 1.6: Concepts du protocole SOA

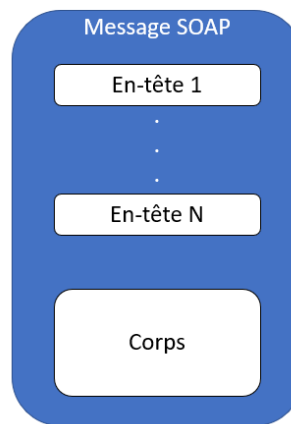


Figure 1.7: Structure d'un message SOAP

5.3. REST

REST est une architecture de services Web qui a été développée à partir de plusieurs styles architecturaux basés sur réseau qui est souvent appliqué dans les services basés sur internet [13]. L'architecture REST est généralement exécutée via HTTP (HyperText Transfer Protocol), impliquant le processus de lecture d'une page Web particulière qui contient un Fichiers XML ou JSON. Chaque demande est indépendante, le serveur ne sauvegarde pas l'état de la requête quoi qu'il en soit. Interface de programmation d'application (API) qui suit un style REST est appelé une API RESTful. Reposant L'API utilise l'URI (Uniform Resource Identifier) pour représenter les ressources. Toute source de données est identifiée à l'aide du lien URI. Quelle méthode utilisée dans REST incluent : GET pour obtenir des ressources, POST est utilisé pour créer une nouvelle ressource et utiliser la méthode PUT pour mettre à jour les ressources par source Puissance. Alors que la méthode DELETE est utilisée pour supprimer une ressource ou un ensemble de ressources. En général, la méthode utilisée dans REST est montrée dans le tableau 1.1.

Resource	Method			
	Get	Post	Put	Delete
/api/alarms	Get a list of all alarm	Create a new list of alarms.	Update a list of alarms	Delete all alarm
/api/alarms/1	Get an alarm by alarm's ID	Treat as a collection. Create a new alarm in it.	If an alarm exists, update the alarm. If alarm does not exist. Create a new alarm	Delete the alarm.

Tableau 1.1 : Exemple de la méthode REST

6. Description

➤ WSDL (Web Service Description Language) :

C'est un schéma en format XML permettant de décrire un service Web en précisant les méthodes disponibles, les formats des messages d'entrée et de sortie et comment y accéder. Il définit un espace de travail extensible pour décrire des interfaces de Services Web. Il a été premièrement développé par Microsoft et IBM et soumis au W3C. Comme les autres technologies XML, WSDL est extensible et assure une compatibilité entre les différentes implémentations. Si l'expéditeur et le destinataire d'un message peuvent partager et comprendre le fichier WSDL de la même façon, alors l'interopérabilité sera assurée. WSDL permet de décrire :

- Types des données ;
- Messages ;
- Opérations ;
- Binding ;

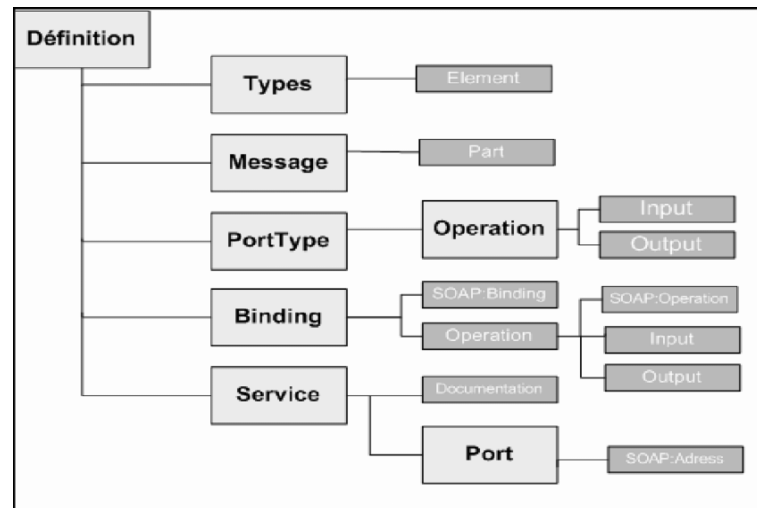


Figure 1.8: Structure d'un document WSDL [12]

7. Publication

➤ UDDI

UDDI est une spécification technique basée sur XML pour décrire, découvrir et intégrer des services Web, permettant leur publication et leur recherche.[14] L'annuaire UDDI est divisé en deux parties principales : une partie publication ou inscription et une partie découverte. La section Publication rassemble toutes les informations relatives à l'entreprise et à ses services, qui sont introduites via l'API d'enregistrement. La partie découverte facilite la recherche d'information contenue dans un UDDI grâce à l'API SOAP. Une publication dans un registre UDDI est un document XML composé de plusieurs éléments, dont les principaux sont :

- **BusinessEntity** : Représente la description de l'organisation (entreprise, serveur, personnes, ...) qui publie le service dans l'annuaire ;
- **BusinessService** : Ensemble d'informations relative aux services web fournis par la businessEntity ;
- **BindingTemplate** : Ensemble de données techniques du service offert, concernant le lieu, le point d'entrée (URL) et le mode d'utilisation de service (protocole d'accès) ;
- **tModel(Technical Model)** : Englobe des informations supplémentaires du service telles que le mode d'accès, les conditions d'utilisation, les garanties, etc ;

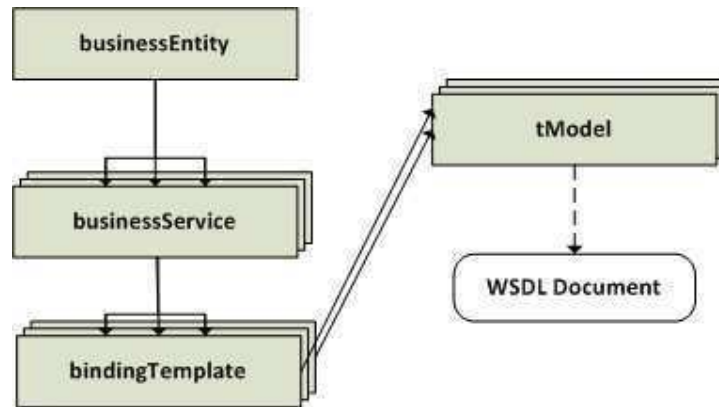


Figure 1.9 : Structure de données au sein de l'UDDI.[15]

Ensemble ces éléments sont répertoriés en trois catégories, pour faciliter l'opération de recherche d'information inclut dans un annuaire UDDI :

- ❖ **Pages blanches** : disposent des informations, associées à l'entreprise offrant le service, qui sont déjà renseignées par le fournisseur dans l'élément Business Entity lors de la publication (le nom de l'entreprise, ses coordonnées, la description de l'entreprise...etc.)
- ❖ **Pages jaunes** : est une description non technique détaillée, du fournisseur de service permettant de répertorier l'entreprise notamment la catégorie, le type de service offert, le secteur d'activité, ect.
- ❖ **Pages vertes** : disposent des informations techniques précises, liées aux services fournis, basées sur leur description WSDL et indiquant les moyens d'y accéder.

Le protocole d'utilisation de l'UDDI, contient trois fonctions de base [16] :

- Publish : Pour enregistrer un nouveau service ;
- Find : Pour interroger l'annuaire ;
- Bind : Pour effectuer la connexion entre l'application client et le service ;

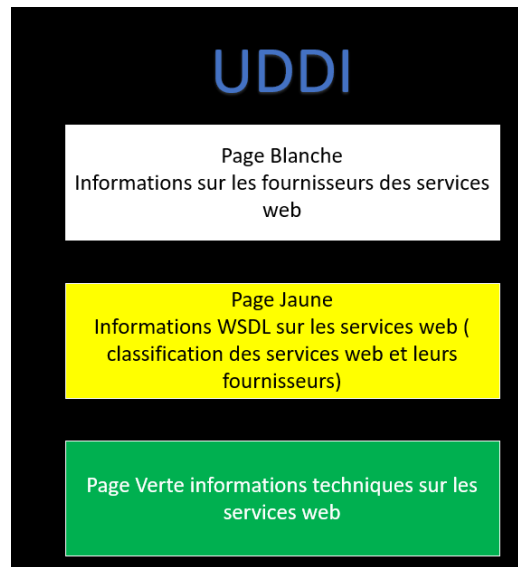


Figure 1.10: Les trois pages de l'annuaire UDDI.[15]

8. Comparaison entre SOAP & REST

Pendant des années, les professionnels de l'informatique et les développeurs Web ont débattu pour savoir quel service Web était le meilleur et pourquoi. Les services SOAP et REST présentent tous les deux des avantages et des désavantages et peuvent être employés de façon complémentaire dans une architecture orientée service, ils offrent différentes variantes. Par exemple, les services Web REST sont souvent un meilleur choix lorsque le temps est un facteur, mais lors de la création de services avec plusieurs méthodes non CRUD (create, read, update, delete), SOAP l'emporte. À moins que le fournisseur WS ne décide à l'avance, les exigences spécifiques à l'entreprise dictent le type de services Web qu'un partenaire mettra en œuvre. Alors parfois les deux options sont la bonne réponse. C'est le cas des deux plus grandes entreprises de commerce électronique au monde : Amazon et eBay, qui utilisent REST et SOAP pour les services Web. Alors que de plus en plus d'organisations se concentrent sur les services et explorent davantage de fonctionnalités, elles n'auront d'autre choix que de prendre en charge les deux types. Outre à partir de là, REST est meilleur que SOAP dans le temps de réponse et des réponses basées sur la taille des données [17], [16], [18], [19].

9. L'intérêt d'un service web

Les technologies des services Web peuvent être appliquées à toute sorte d'applications auxquelles elles offrent de considérables avantages en comparaison aux ancienne API propriétaires, aux implémentations spécifiques à une plate-forme et à quelques autres

restrictions classiques que l'on peut rencontrer (multi plate-forme, multi langage, disponible sur l'internet avec une information actualisée disponible en temps réel, ...).

Les entreprises qui mettent à disposition leurs services Web permettent aux développeurs intéressés par ses fonctionnalités de les réutiliser sans avoir à les recoder. Le principe des services Web permet d'avoir un partage des fonctionnalités et facilite grandement le développement, leur internet majeur et l'instauration :

- L'interopérabilité des applications
- L'intégration des applications sur le Web.

10. Caractéristiques des Services web

Les services web se caractérisent par les caractéristiques suivantes :

- **L'interopérabilité** : C'est la capacité des services web d'interagir avec d'autres composantes logicielles via des éléments XML et utilisant des protocoles de l'Internet.
- **La simplicité** : Les services web réduisent la complexité des branchements entre les participants. Cela se fait en ne créant la fonctionnalité qu'une seule fois plutôt qu'en obligeant tous les fournisseurs à reproduire la même fonctionnalité à chacun des clients selon le protocole de communication supporté.
- **Une composante logicielle légèrement couplée** : L'architecture modulaire des services Web, combinée au faible couplage des interfaces associées, permet l'utilisation et la réutilisation de services qui peuvent facilement être recombinaés à différentes autres applications.
- **L'hétérogénéité** : Les services web permettent d'ignorer l'hétérogénéité entre les différentes applications. En effet, ils décrivent comment transmettre un message (standardisé) entre deux applications, sans imposer comment construire ce message.
- **Auto-descriptivité** : Les services web ont la particularité d'être autodéscriptifs, c'est à dire capables de fournir des informations permettant de comprendre comment les manipuler. La capacité des services à se décrire par eux-mêmes permet d'envisager l'automatisation de l'intégration de services.

11. Conclusion

Les services Web sont l'un des éléments clés de ce que l'on appelle le Web programmable. Il s'agit d'éléments logiciels extrêmement polyvalents qui ont réellement le potentiel d'ouvrir une nouvelle ère dans le domaine des logiciels : l'ère de l'interopérabilité. Les services Web peuvent être utilisés efficacement pour participer à des transactions interentreprises (B2B) et les mettre en place. Ils permettent d'exposer les fonctionnalités des logiciels aux clients et d'intégrer des plateformes hétérogènes.

Dans ce chapitre, nous introduisons d'abord le service web, son Fonctionnement et démarche côté client ainsi que coté serveur, ensuite nous avons énuméré les avantages et les inconvénients des services web. Puis, on a détaillé le concept de service web avec l'architecture de référence de ce dernier, ensuite nous avons présenté les technologies sur lesquelles les services web sont basées tels que XML, SOAP, REST et WSDL, UDDI. Et enfin nous avons parlé de l'intérêt d'un service web et ses caractéristiques. Dans le chapitre suivant nous allons présenter la gestion des alarmes et les types d'alarmes DCS , ainsi que la chaudière industrielle et son fonctionnement.

Gestion des alarmes

1. Introduction

La gestion des alarmes fait référence à la conception, à la mise en œuvre, à l'exploitation et à la maintenance efficaces des alarmes des usines de fabrication industrielle/de traitement. La gestion des alarmes est nécessaire dans un environnement d'usine de traitement contrôlé par un opérateur utilisant un système de contrôle, tel qu'un DCS ou un contrôleur logique programmable (PLC). Les alarmes sont déclenchées lorsque la valeur du processus s'écarte des conditions de fonctionnement normales, c'est-à-dire lors de l'apparition de situations de fonctionnement anormales. Une situation anormale se produit lorsqu'une perturbation dans un processus fait dévier le fonctionnement de l'usine de son état de fonctionnement normal, ce qui nécessite une intervention humaine. L'incapacité à diagnostiquer et à contrôler les situations anormales a un impact de plusieurs milliards de dollars sur l'économie. La gestion des alarmes vise essentiellement à prévenir, ou du moins à minimiser, les pertes physiques et économiques par l'intervention de l'opérateur en réponse à la condition qui a fait l'objet de l'alarme. La gestion des alarmes fait référence aux processus et pratiques de détermination, de documentation, de conception, de surveillance et de maintenance des messages d'alarme provenant des systèmes d'automatisation des processus et de sécurité. Un système avancé de gestion des alarmes est une combinaison d'applications qui permettent de créer des systèmes d'alarme efficaces, conformes aux recommandations et aux normes industrielles telles que la publication 191 de l'Engineering Equipment & Materials User's Association (EEMUA), la norme 18.2 de l'International Society of Automation (ISA) et les meilleures pratiques du Abnormal Situation Management (ASM) Consortium. Dans cette partie du chapitre on va élaborer les généralités sur les alarmes, présenter les différents types et discuter les concepts de leur gestion [20].

2. Définition d'une alarme

L'American National Standards Institute (ANSI) et l'International Society of Automation (ISA) ont collaboré pour définir la notion d'alarme, ainsi que la méthode de gestion des alarmes jugée optimale, en créant la norme ANSI/ISA 18.2 (International Society of Automation 2009). Une alarme est un moyen audible et/ou visible d'indiquer à l'opérateur un dysfonctionnement

de l'équipement, une déviation du processus ou une condition anormale nécessitant une intervention. Chaque alarme nécessite une réponse d'un opérateur, il est important que ce dernier ait le temps d'analyser chaque alarme, ainsi que ses causes et ses conséquences, pour comprendre et résoudre le problème ayant généré cette alarme. Il est donc primordial que les opérateurs ne soient pas débordés par le nombre d'alarmes pour qu'ils aient le temps d'accomplir leur mission correctement. Ainsi, si aucune action de l'opérateur n'est nécessaire, le signal ne doit pas être déclaré en tant qu'alarme, mais en tant que simple événement informatif de l'état du système. Il est donc nécessaire de s'interroger dès la création du système d'alarme sur la nature des informations transmises à l'opérateur et notamment sur la nécessité de les qualifier en tant qu'alarme. [21]

3. Les types d'alarmes DCS

3.1 Alarme de procédé

Ces alarmes peuvent concerner l'efficacité du processus ou indiquer des défauts dans l'équipement. Ce type d'alarme est normalement intégré au système de contrôle de l'usine (généralement un DCS) et partage les mêmes capteurs que le système de contrôle. [23]

3.2 Alarmes liées à la sécurité

Ces alarmes sont utilisées pour alerter les opérateurs d'une condition qui peut être potentiellement dangereuse ou dommageable pour l'installation. Ces alarmes doivent normalement avoir une priorité élevée et lorsqu'elles sont impliquées dans la protection contre un mauvais fonctionnement par le système de contrôle, elles doivent être indépendantes des dispositifs qu'elles surveillent. Dans de nombreux cas, ces alarmes sont générées par le système d'arrêt de sécurité lui-même. [23]

3.3 Alarme d'arrêt d'urgence

Ce type d'alarme indique à l'opérateur qu'un événement d'arrêt automatique a été atteint et a été déclenché par le SIS. Cependant, il faut se rappeler que tous les systèmes d'alarme n'entraînent pas nécessairement des arrêts. Dans certains cas, ces alarmes sont le résultat d'un démarrage automatique. [23]

4. Cycle de vie de la gestion des alarmes

Le modèle de cycle de vie est utile pour identifier les exigences et les responsabilités pour la mise en œuvre d'un système de gestion des alarmes. Il est aussi applicable à l'installation de nouveaux systèmes d'alarme ou à la gestion d'un système existant.

La figure 2.1 illustre la relation entre les étapes du cycle de vie de la gestion des alarmes décrites dans cette norme. Le cycle de vie de la gestion des alarmes couvre les activités de spécification, de conception, de mise en œuvre, d'exploitation, de surveillance, de maintenance et de changement du système d'alarme, de la conception initiale à la mise hors service. de la conception initiale à la mise hors service.

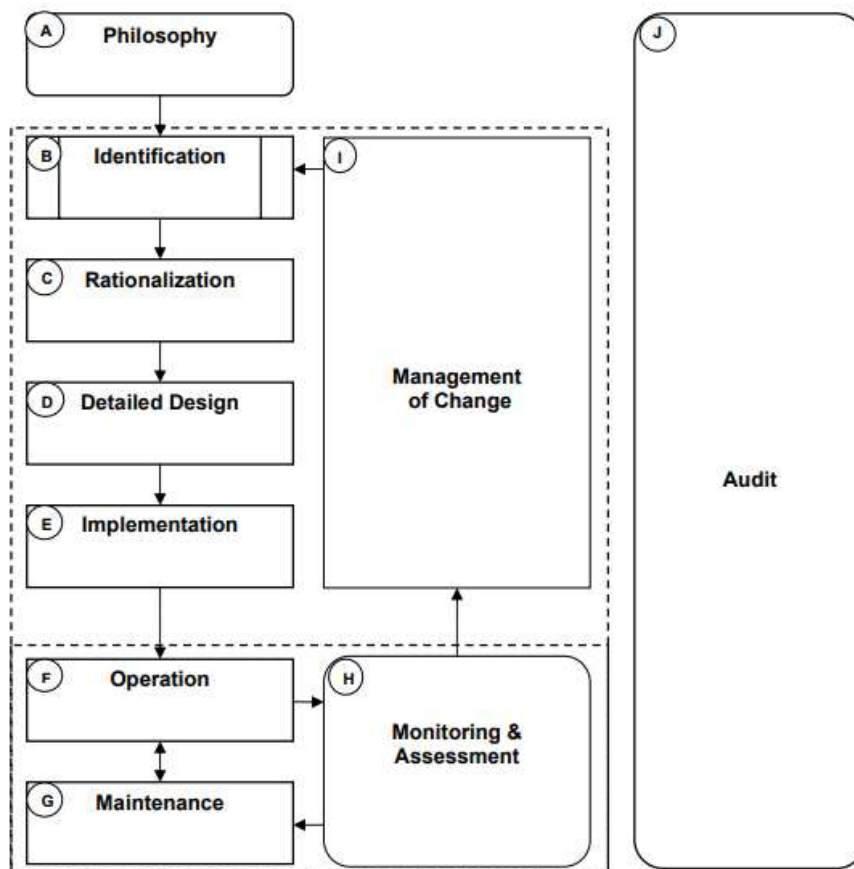


Figure 2.1: Alarm Management Life cycle [21]

Nous allons maintenant décrire les étapes du cycle de vie de la gestion des alarmes présentées dans la figure 2.1 :

4.1 Alarm Philosophy (A)

Une planification de base est nécessaire avant de concevoir un nouveau système d'alarme ou de modification un système existant. En général, la 1ere étape est le développement d'une philosophie d'alarme qui part des définitions de base et les étend aux définitions opérationnelles.

La définition des priorités d'alarme, des classes, des mesures de performance, des limites de performance et des exigences en matière de rapports est déterminée en fonction des objectifs, des définitions et des principes. Les schémas de présentation des indications d'alarme dans l'IHM, y compris l'utilisation des priorités, sont également définis dans la philosophie d'alarme, qui doit être cohérente avec la conception globale de l'IHM. [22]

4.2 Identification (B)

L'étape d'identification est un point de collecte des alarmes potentielles proposées par l'une des nombreuses méthodes permettant de déterminer qu'une alarme peut être nécessaire. Les méthodes peuvent être formelles, telles que l'analyse des dangers du processus, les spécifications des exigences de sécurité, les recommandations d'une enquête sur un incident, les bonnes pratiques de fabrication, les permis environnementaux, le développement de P&ID ou les examens des procédures d'exploitation. Les modifications du processus et les tests de fonctionnement peuvent également générer le besoin d'alarmes ou de modifications. [3]

4.3 Rationalization (C)

L'étape de rationalisation réconcilie le besoin identifié d'une alarme ou d'un changement de système d'alarme avec les principes de la philosophie d'alarme. La rationalisation est le processus qui consiste à appliquer les exigences relatives à une alarme et à générer la documentation correspondante, comme la base du point de consigne de l'alarme, la conséquence et les mesures correctives que peut prendre l'opérateur. Le type de conséquences d'une alarme rationalisée ou d'autres critères, peuvent être utilisés pour séparer les alarmes en classes telles que définies dans la philosophie de l'alarme.

Les résultats de la rationalisation sont documentés, généralement dans la base de données principale des alarmes (c'est-à-dire dans un document ou un dossier approuvé), qui est conservée pendant toute la durée de vie du système d'alarme. [22]

4.4 Detailed Design (D)

Au stade de la conception, les attributs d'alarme sont spécifiés et conçus sur la base des exigences déterminées par la rationalisation. Il y a trois domaines de conception : la conception de l'alarme de base, la conception de l'IHM et la conception des techniques d'alarme avancées. La conception de base de chaque alarme suit des conseils basés sur le type d'alarme et le système de contrôle spécifique. La conception de l'IHM comprend l'affichage et l'annonce des alarmes, y compris les indications de priorité des alarmes. Les techniques d'alarme avancées sont des fonctions supplémentaires qui améliorent l'efficacité du système d'alarme au-delà de l'alarme de base et de la conception de l'IHM. Ces méthodes comprennent l'alarme basée sur l'état et la priorisation dynamique. [22]

4.5 Implementation (E)

Au stade de la mise en œuvre, les activités nécessaires pour installer une alarme ou un système d'alarme et le mettre en état de fonctionnement sont achevées. La mise en œuvre d'une nouvelle alarme ou d'un nouveau système d'alarme comprend l'installation physique et logique et la vérification fonctionnelle du système. Les opérateurs étant une partie essentielle du système d'alarme, leur formation est une activité importante lors de la mise en œuvre. Le test des nouvelles alarmes est souvent une exigence de mise en œuvre. La documentation relative à la formation, aux tests et à la mise en service peut varier selon la classification définie dans la philosophie de l'alarme. [22]

4.6 Operation (F)

Au cours de la phase d'exploitation, l'alarme ou le système d'alarme est actif et remplit la fonction pour laquelle il a été conçu. Une formation de remise à niveau sur la philosophie de l'alarme et le but de chaque alarme est incluse dans cette étape. [22]

4.7 Maintenance (G)

Au stade de la maintenance, l'alarme ou le système d'alarme n'est pas opérationnel mais est testé ou réparé. Une maintenance périodique (par exemple, le test des instruments) est nécessaire pour s'assurer que le système d'alarme fonctionne comme prévu. [22]

4.8 Monitoring and Assessment (H)

Au cours de la phase de surveillance et d'évaluation, la performance globale du système d'alarme et les alarmes individuelles sont surveillées en permanence par rapport aux objectifs

de performance énoncés dans la philosophie d'alarme. Le suivi et l'évaluation des données de la phase d'exploitation peuvent déclencher des travaux de maintenance ou identifier la nécessité de modifier le système d'alarme ou les procédures d'exploitation. La surveillance et l'évaluation des données de la phase de maintenance fournissent une indication de l'efficacité de la maintenance. Les performances globales du système d'alarme sont également contrôlées et évaluées par rapport aux objectifs de la philosophie d'alarme. Sans surveillance, un système d'alarme est susceptible de se dégrader. [22]

4.9 Management of Change (I)

Dans la phase de gestion du changement, les modifications du système d'alarme sont proposées et approuvées. Le processus de changement doit suivre chacune des étapes du cycle de vie, de l'identification à la mise en œuvre. La mise en œuvre. [22]

4.10 Audit (J)

Au stade de l'audit, des examens périodiques sont effectués pour maintenir l'intégrité du système d'alarme et des processus de gestion des alarmes. Les audits de la performance du système peuvent révéler des lacunes qui ne sont pas apparentes dans la surveillance de routine. L'exécution par rapport à la philosophie d'alarme est auditée pour identifier les améliorations du système, telles que les modifications de la philosophie d'alarme. Les audits peuvent également identifier le besoin de renforcer la discipline de l'organisation pour suivre la philosophie d'alarme. [22]

5. Gestion des alarmes

La gestion des alarmes est l'application des facteurs humains (ou "ergonomie") ainsi que l'ingénierie de l'instrumentation pour gérer la conception d'un système d'alarme afin d'en améliorer la convivialité. Le plus souvent, le principal problème d'utilisabilité réside dans le fait qu'un trop grand nombre d'alarmes sont annoncées lors d'une perturbation de l'installation, communément appelée inondation d'alarme. Dans les activités quotidiennes en salle de contrôle, les opérateurs rencontrent fréquemment les problèmes suivants :

- Surcharge d'alarmes : Plus d'alarmes qu'il n'est possible de gérer ;
- Inondations d'alarme ;
- Alarmes intempestives ;
- Les alarmes de type statique ;
- Alarmes redondantes ;

- Alarmes qui n'ont pas de réponse ;
- Alarmes avec une mauvaise priorité ;

6. La chaudière industrielle (cas d'étude)

Nous consacrons cette partie à l'étude d'une chaudière Industrielle installée dans le complexe de raffinage situé à Skikda (RA1K). La chaudière est un dispositif permettant de chauffer l'eau et de produire de la vapeur. On utilise les chaudières pour produire la vapeur nécessaire au fonctionnement des procédés par exemple : des turbopompes, des turbocompresseurs, des turbos-ventilateurs. La source de chaleur peut être fournie par un combustible (gaz, fioul, charbon...) ou une résistance électrique. Dans la nouvelle industrie, les chaudières occupent une place importante dans la plupart des applications industrielles et représente un critère important pour le raffinage du pétrole. [25].

6.1 Définition d'une chaudière

Une chaudière industrielle peut être définie comme un système dans lequel l'énergie d'un combustible (entrée) est cédée et transférée à un liquide pris dans le cas des chaudières à vapeur, L'énergie produite permet de changer l'état d'une phase liquide à une phase gazeuse (sortie). Le fluide chauffé est transporté à l'aide de tuyauteries jusqu'au lieu d'utilisation. Généralement une chaudière fonctionne en circuit fermé, le fluide chaud sort par une tuyauterie fixée sur sa partie haute et revient par une autre tuyauterie fixée sur sa partie basse après avoir circulé et s'être refroidi. [24].

6.2 Principaux éléments d'une chaudière

La chaudière est une machine complexe constituée d'équipement et d'instruments présentés dans le P&ID de la chaudière installée dans le complexe RA1K (raffinerie de Skikda) :

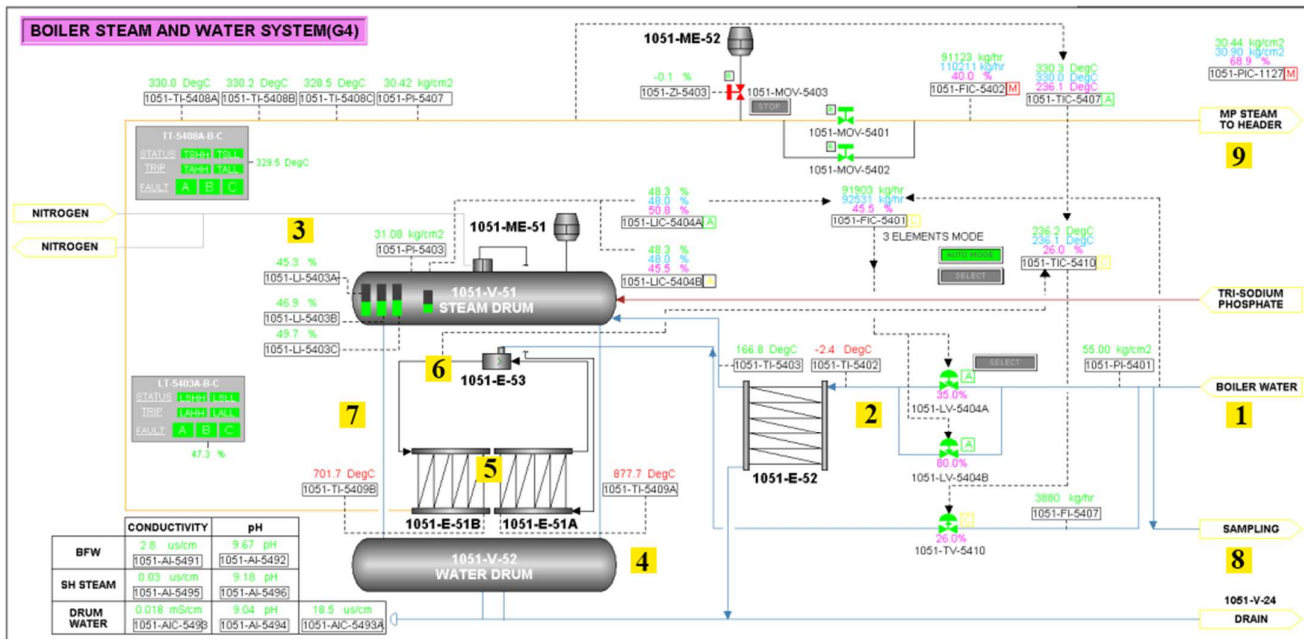


Figure 2.2: Diagramme P&ID de la chaudière

- 1: Pompe d'alimentation
- 2: Economiseur
- 3: Ballon supérieur
- 4: Ballon inférieur
- 5: Surchauffeur
- 6: Désurchauffeur
- 7: Chambre de combustion
- 8: Système d'échantillonnage
- 9: Vapeur produite

6.2.1. Les équipements :

La chaudière se compose de plusieurs équipements, on va citer les équipements les plus importants :

L'économiseur extrait des gaz d'échappement de la chaudière, la chaleur en excès, que la chaudière elle-même ne peut plus absorber. L'économiseur est généralement constitué de tubes à ailettes ou de tubes lisses. Pour préchauffer l'eau qui est débité par la suite par la pompe alimentaire, qui est acheminée vers le ballon supérieur. Après cela la chaleur récupérée par l'eau d'alimentation dans l'économiseur est restituée. [26]



(a) Economiseur à ailettes

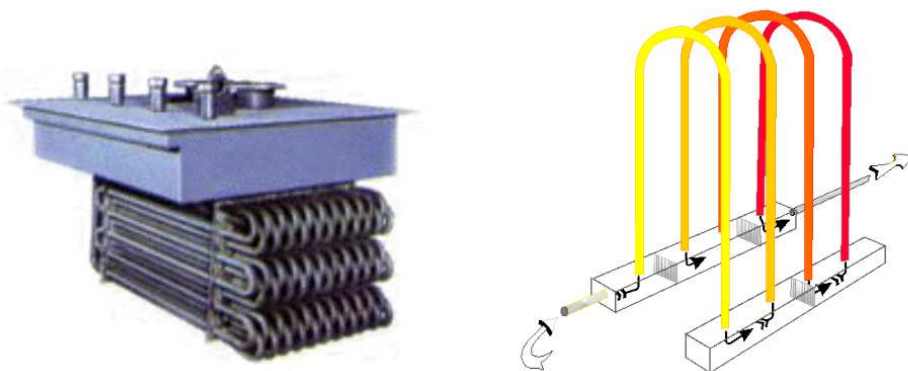


(b) Economiseur lisse

Figure 2.3: Economiseurs de la chaudière

Il y a deux **surchauffeurs**, un surchauffeur primaire travaillant à haute température (SHT), et un secondaire qui travaille à basse température (SBT), connectés en série, placés dans la partie haute du puit arrière du générateur de vapeur. Ils fonctionnent pour augmenter la température de la vapeur saturée sortant du réservoir jusqu'au 487 °C (vapeur surchauffée).

Les surchauffeurs reçoivent la chaleur par convection des fumées qui sort de la chambre de combustion. La variation de la température de la vapeur à travers les surchauffeurs est représentée sur la figure 2.4. La température de la vapeur surchauffée est contrôlée par une injection d'eau de désurchauffe pratiquée au niveau du collecteur inter surchauffeurs. Deux vannes de désurchauffe, aux caractéristiques identiques (un débit d'eau réglable de 0.5 t/h à 25 t/h avec une différence de pression de 20 à 16 bars) travaillent en cascade.

**Figure 2.4:** présentation du Surchauffeur

Les désurchauffeurs sont destinés à refroidir et moduler la température de la vapeur surchauffée ou resurchauffée, ils sont rarement tubulaires mais généralement assimilables à des échangeurs par mélange puisqu'ils procèdent par injection d'eau dans la vapeur. [26]

Les resurchauffeurs ont un rôle identique à celui des surchauffeurs, à savoir élever la température de la vapeur resurchauffée provenant de la turbine.

Le ballon supérieur se situe à la sortie de l'économiseur, l'eau d'alimentation monte vers un réservoir situé dans la partie supérieure de la chaudière. Et vers lequel se collecte le mélange eau vapeur provenant des écrans. C'est à son niveau que s'effectue la séparation entre la phase liquide et la phase vapeur. Le DCS se charge de maintenir à 0 le niveau d'eau dans le ballon, ce niveau ne doit strictement pas dépasser $\pm 150\text{mm}$. [29].

Le ballon inférieur est un autre réservoir en bas de la chambre de combustion. Il joue le rôle de collecteur d'eau. Il est toujours plein pour des raisons de sécurité de la chaudière, afin de pouvoir évacuer de l'eau à travers le tube porteur vers les écrans au cas de niveau bas.

La chambre de combustion est constituée par des tubes. C'est à ce niveau que se fait la combustion. Le plancher est protégé par des briques réfractaires qui aident le foyer à se maintenir à une température très élevée. Nous devons aussi assurer l'introduction du combustible et de l'air en proportion convenable et leur mélange en vue d'obtenir une combustion complète et rationnelle [26]

Les pompes alimentaires sont des pompes qui changent l'énergie mécanique en énergie hydraulique. Elles sont les plus utilisées dans les installations industrielles. Leur popularité est due à la simplicité de construction, à leurs petites exigences d'espace et à leur capacité d'opération à hautes vitesses. La pompe alimentaire sert à alimenter la chaudière en eau déminéralisée. Dans l'installation, il y a deux pompes alimentaires, une en marche, l'autre en arrêt. Ces dernières sont des pompes centrifuges horizontales de type Les vannes et les clapets DVMX 6x8x11 E/4 étages.

6.2.2 Les instruments :

Un capteur de niveau est un dispositif électronique qui permet de mesurer la hauteur du matériau, en général du liquide, dans un réservoir ou un autre récipient.

Tous les systèmes pétrochimiques utilisent ces capteurs pour mesurer le niveau de fluide. Il convient de connaître la nature de fluide, leur propriété physique et chimique, la distance et conditions de sécurité. [27].

Les capteurs de débit sont des capteurs utilisés pour la mesure de débit. Ils utilisent des techniques qui peuvent être mises en œuvre. On trouve essentiellement ce type dans les organes déprimogènes : diaphragme, venturi, tuyère.

Les transmetteurs de pression peuvent être électroniques intelligents, analogiques ou basse tension. Ces capteurs existent dans une gamme très variée de configurations destinée aux applications de mesure de la pression différentielle. Ont des cartes de circuit imprimé enfichables à base de circuits intégrés.

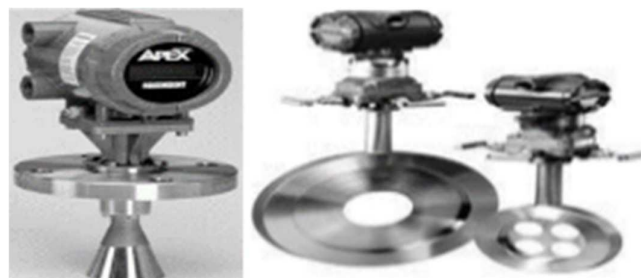
Les deux **capteurs de températures** les plus utilisés actuellement pour la télémessure sont :

Les thermocouples qui sont dans un circuit fermé constitué de deux conducteurs de nature différente, il circule un courant lorsqu'on maintient entre les deux jonctions une différence de température.

La sonde est essentiellement utilisée pour les mesures continues et détections de niveau dans les liquides.



(a) Thermocouple (b) Sonde (c) indicateur de pression



(d) Capteur de niveau (e) indicateur de débit.

Figure 2.5: Les instruments de la chaudière

7. Description fonctionnelle

Les logiques décrites font référence aux boucles de commande indiquées dans le P&ID et doivent être mises en œuvre dans le système de commande distribué (DCS). Celles-ci sont partiellement et uniquement pertinentes pour le contrôle et l'automatisation indiqués dans le P&ID. Elle doit être intégrée, interfacée et harmonisée avec le reste de l'installation avec la logique BOP, au moyen de la coordination du contrôle de la philosophie générale de l'installation. On va décrire les fonctions, les logiques et les interfaces des systèmes de contrôle des chaudières à mettre en œuvre dans un système de contrôle distribué (DCS) pour cinq chaudières à vapeur identiques installées à la raffinerie de Skikda - Algérie. Le système DCS est le système de contrôle dédié au contrôle et à la supervision de toutes les activités de la chaudière (à l'exclusion de celles relatives aux sécurités de combustion et aux logiques de brûleurs contrôlés par le BMS) et à l'assistance de l'opérateur pendant le démarrage, le fonctionnement et l'arrêt de la chaudière. Le DCS reçoit et envoie des signaux de commande et d'information à partir de/vers le BMS et l'UPLC (Plc) pour contrôler la logique de démarrage/arrêt des deux ventilateurs d'air de combustion, et le ventilateur du scanner UV de refroidissement/nettoyage et ses accessoires. Le système DCS permettra des opérations manuelles ou automatiques de toutes les boucles de contrôle des chaudières et surveillera également en permanence toutes les opérations des brûleurs et les conditions d'utilisation du combustible [28].

8. Valeur de conception de référence

Les conditions environnementales dans lesquels la raffinerie de Skikda travail :

- Site : Skikda (Algérie) ;
- Élévation : 13 m. a.s.l ;
- Température normale : +15 ÷ +35 °C ;
- Température maximale de conception : +47 °C ;
- Température minimale de conception : + 1 °C ;
- Humidité relative normal : 72% ;
- Humidité relative maximale : 79% ;
- Zone sismique : ;
- Zone dangereuse : Classé Zone 2 G [28].
- Atmosphère : Saline, corrosif pétrochimique ;

9. Description de l'architecture des systèmes

Les systèmes suivants sont impliqués dans le fonctionnement de la chaudière :

- Système de gestion des brûleurs (BMS) ;
- Système de contrôle distribué (DCS) ;
- PLC pour l'automatisation des ventilateurs d'air de combustion et des ventilateurs de refroidissement/nettoyage des scanners UV (UPLC) ;
- Système d'arrêt d'urgence (ESD) Si présent [28].

Le BMS assurera les fonctions suivantes

- Superviser la protection de la chaudière, en déclenchant automatiquement la chaudière en cas d'urgence par les actions nécessaires sur les brûleurs ;
- Mesurer ou détecter les conditions du processus de sécurité [28].

Le DCS assurera les fonctions suivantes :

- Superviser le démarrage et l'arrêt de la chaudière ;
- Mesurer ou détecter les conditions du processus ;
- Afficher les informations de statut provenant du BMS, de l'UPLC et du MCC ;
- Calculer automatiquement les ajustements nécessaires, en fonction des écarts entre les conditions du processus et les points de consigne, et en enregistrant les tendances des variables du processus ;
- Sortir un signal pour contrôler les éléments finaux dans le but de contrôler le processus [28].

10. Conclusion

Dans ce chapitre nous avons présenté les alarmes et leurs différents types, nous avons également présenté la chaudière industrielle qui est notre cas d'études. Dans le chapitre suivant nous allons élaborer la conception et la réalisation de notre système en présentant les modèles nécessaires

Conception

1. Introduction

Les usines industrielles ne cessent de grandir et se développer. Au fur et à mesure que la taille de ces processus industriels augmente, le volume des informations d'alarmes présentées aux opérateurs augmente également. Dans ce chapitre, nous allons étudier les ordres temporels et les intervalles entre les événements afin d'extraire de manière efficace et significative des modèles à partir d'une séquence d'événements d'alarme.

Notre principale contribution est la démonstration d'une stratégie de segmentation et de filtrage basée sur les événements tout en préservant la richesse des informations d'alarme.

2. Objectif du travail

L'objectif principal est de créer un service web pour l'aide à la décision. Nous proposons une approche Data Mining afin de découvrir les corrélations entre les séquences d'alarmes. Il s'agit de créer un modèle à partir d'un historique d'alarmes récupéré lors d'un stage pratique à Sonatrach. A partir de ces objectifs nous traçons les grands axes à développer :

- Description de l'architecture globale de notre système
- Création d'un modèle d'apprentissage automatique.
- Création d'un service web pour supporter ce modèle d'apprentissage.

3. Travaux existants

Récemment, les chercheurs universitaires et les ingénieurs industriels ont accordé une attention croissante aux corrélations entre les alarmes. Ils ont présenté une procédure basée sur l'analyse de la corrélation entre les variables physiques et les variables d'alarme. Cette procédure pourrait suggérer des paramètres d'alarme optimaux afin de faciliter le processus de rationalisation des alarmes. D'autres facteurs, tels que les délais de détection et la priorisation des alarmes, ont été discutés pour rendre la conception plus fiable et précise [30]. Noda et al. [31] ont proposé un schéma appelé analyse de corrélation d'événements pour identifier les alarmes corrélées et les actions de fonctionnement inutiles. Gao et al. [32] ont proposé le modèle structurel interprétatif spatial Novel pour représenter l'organisation hiérarchique de l'espace et montrer les relations causales à différents niveaux de granularité.

4. Architecture du système

L'architecture de référence de notre service web, illustrée dans la figure 3.1, s'articule autour:

- **Le fournisseur de service** : correspond au propriétaire du service. D'un point de vue technique, il est constitué par la plate-forme d'accueil du service. Les infrastructures logicielles (Framework) utilisées pour le fournisseur de service sont python web Framework FLASK avec lequel on a programmé notre API REST.
- **Le client** : correspond au demandeur de service. D'un point de vue technique, il est constitué de l'application qui va rechercher et invoquer un service. Les infrastructures logicielles (Framework) lesquels on a utilisé pour tout notre FRONT-END sont REACTJS et ELETRONJS qui utilisent Javascript.

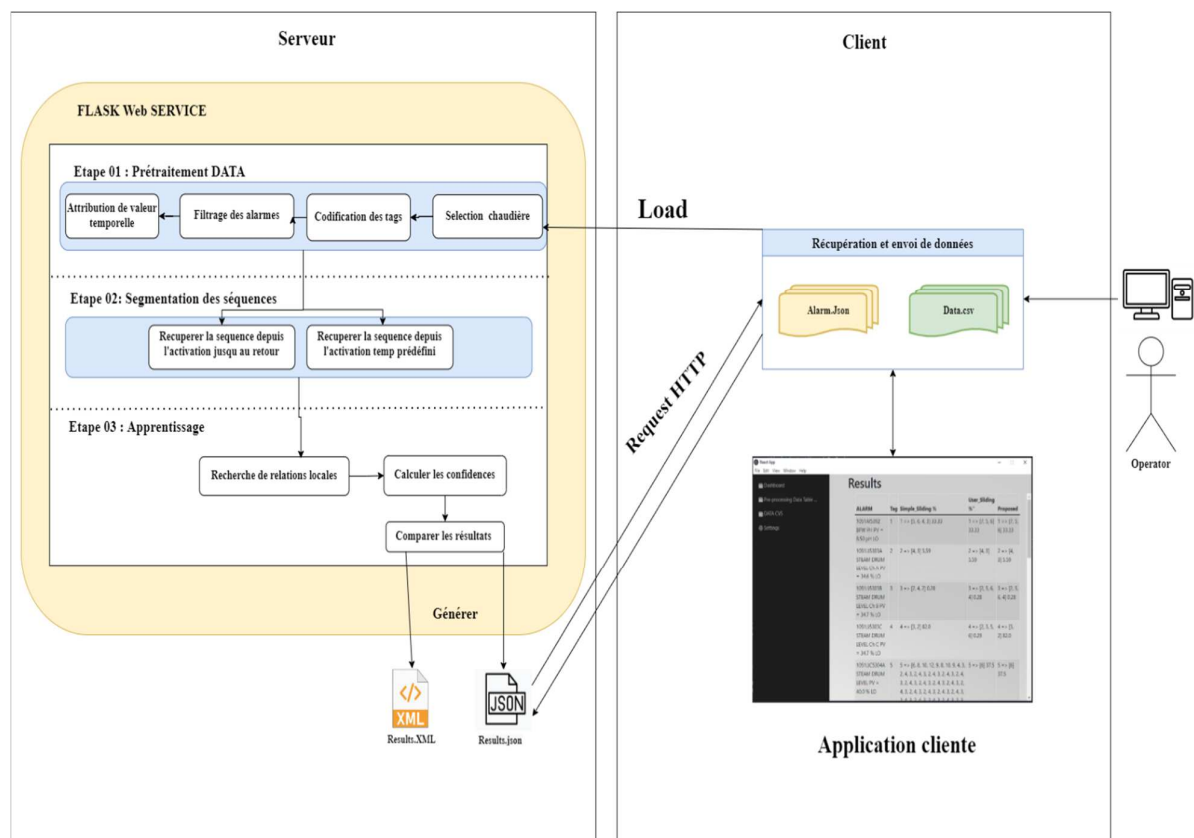


Figure 3.1: Architecture du système.

4.1. Interaction Client- Serveur

Les interactions de base entre ces deux rôles incluent les opérations de publication, de recherche et de liens d'opérations (figure 3.2). Un scénario type d'utilisation de cette architecture peut se décrire comme suit: Le client fait la recherche puis il sélectionne un service donné. Il reçoit ensuite, grâce à la méthode POST, le service sélectionné pour récupérer les informations nécessaires lui permettant de se connecter au fournisseur du service et d'interagir avec l'implémentation du service considéré.

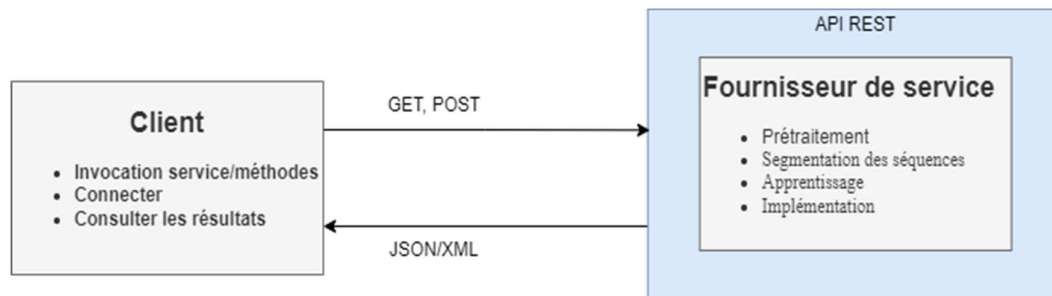


Figure 3.2 : architecture des services web

4.2. Prétraitement

Les bases de données DCS dans les usines contiennent des enregistrements de journaux d'événements horodatés. Comme exigence minimale, chaque enregistrement contient des champs qui stockent des informations telles que l'identifiant unique pour chaque étiquette d'alarme, l'heure et la date, les priorités d'alarme, la description d'alarme qui inclut l'alarme d'activation [ALM] et le retour [RTN]. Notre fichier contient aussi des alarmes systèmes, les modes opératoires, des alarmes redondantes...etc.

L'algorithme 1 décrit les actions appliquées sur le journal d'alarme d'entrée. Ces actions se résument comme suit :

- Cibler la chaudière selon le fonctionnement et la charge;
- Eliminer les alarmes systèmes;
- Eliminer les informations sur les modes opératoires;
- Eliminer les alarmes redondantes qui s'activent deux fois consécutifs sans avoir un retour;
- Codification des tags pour faciliter la mise en œuvre de l'algorithme, chaque Tag a un NumTag unique;

- Discerner l'état d'alarme, quand on a une activation d'alarme $ALM = 1$, et quand on a le retour de cette même alarme $NR = -1$;
- Une séquence d'alarmes est créée, dans laquelle chaque événement est associé à un temps d'occurrence donné sous la forme d'un nombre entier ;

Algorithme 1 : prétraitement

Input : Historique Alarme.csv

Output : PreprocessedDataFramedf

Début

```
df<- lire(Historique_D'alarmes.csv)
```

Codification de tag des alarmes de 1 à N

```
// initialisation de la boucle
```

```
pour i<-0 à n
```

```
Si alarme message égale à ALM
```

```
    Df[i,Alarm] <- Df[I,tag]*1
```

```
Sinon Si alarm message égale a NR
```

```
    Df[i,Alarm] <- Df[I,tag]*-1
```

```
Sinon
```

```
    // Suppression des alarmessystèmes , modes opératoires des alarmes
```

```
    supprimer l'alarme
```

```
fin si
```

```
fin pour
```

```
    // Attribution d'un temps
```

```
Insertion d'unecolonne de temp (s) df['Time']
```

Output df

Fin

4.3. Attribution des valeurs temporelles et segmentation des séquences

Dans notre algorithme nous nous concentrons sur l'activation/retour glissant et le délai limité. Dans la méthode d'activation/retour glissant présenté dans la figure 3.3, une alarme est activée puis nous comptons jusqu'au retour de la même alarme. Nous comptons ainsi tous les événements entre l'activation/retour de cette alarme. Dans la méthode de délai, nous commençons l'activation d'une alarme et comptons jusqu'au délai $<$ délai glissant. Nous comptons aussi tous les événements dans cet intervalle.

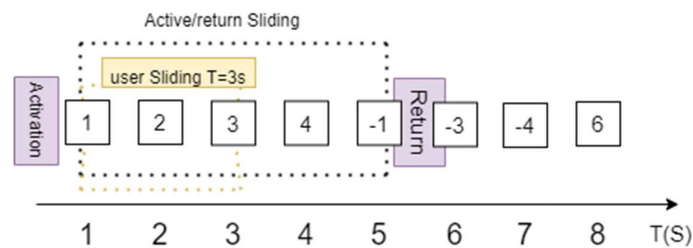


Figure 3.3: Exemple des méthodes de segmentation

L'algorithme 1 montre la première méthode (activation/retour). Nous considérons en entrée le résultat du prétraitement des données et une alarme A. Nous prenons les événements entre chaque activation et retour de cette alarme et nous les stockons dans une liste de transaction E.

L'algorithme 2 montre la deuxième méthode (délai). Nous prenons en entrée le résultat du prétraitement des données, le délai T et une alarme A. Nous comptons les événements entre l'activation et le délai donné puis nous les stockons dans la liste de transaction E1.

Algorithme 1

Input:
 Alarme_code,
 Événement a,
 nombre d'événements N
 Output : List des fenêtres
 transactionnelles E
 Initialiser L<- [], Df<- []
 Changer = faux
 pour i<-1 à n
 Lire(tag)
 Si tag égale a
 changer<-vrai
 Sinon Si s égale -a
 changer<-faux
 finSi
 Si tag != a ou -a
 L<-L+[tag à -tag]
 Sinon
 Df<- E+L
 L<- []
 changer <- faux
 Fin Si
 Fin pour
 Output Df

Algorithme 2

Input:
 Alarme_code,
 Un temps T(s) ,
 Événement a,
 nombre d'événements N
 Output : List des fenêtres
 transactionnelles E
 Initialiser L<- [], Df<- [],t1<-0
 Changer = faux
 pour i<-1 à n
 Lire(tag)
 Si tag égale a
 changer<-vrai
 Sinon Si s égale -a ou t1>=T
 changer<-faux
 T1<-0
 finSi
 Si tag != a ou -a et t1 >=t
 L<-L+[tag à -tag]
 Sinon
 Df<- E+L
 L<- []
 changer <- faux
 Fin Si
 Fin pour
 Output Df

4.4.Apprentissage

Dans cette phase on va récupérer les séquences qui ont été segmenté puis on va calculer le nombre d'occurrence de chaque séquence pour donner le taux d'occurrence.

Algorithme 3 : Apprentissage

Input: Résultat de l'apprentissage

Output Df (sortie de l'algorithme d'apprentissage),

Alarme_Tag

Output T

Initialiser T <- [],

Conf <- 0 ,

Freq<- 0

//Calculer combien de fois Alarme_Tag se produit dans df

Freq<- compter(Alarme_Tag)

pour i dans Df

conf<- (compter i)dans DF/freq)%

T<-(i , conf)

fin pour

output T

5. Etude de cas

5.1. Prétraitement

Dans ce modèle d'apprentissage automatique, nous allons analyser les données historiques récupérées de la chaudière 1051 de la raffinerie de Sonatrach Skikda. Pour cela une étape de prétraitement est nécessaire pour éliminer toutes les données parasites énumérés ci-dessus. Dans notre cas, le fichier d'historique d'alarme comportait 4 chaudières. On cible la chaudière 3.

La figure 3.4 suivante montre le journal d'historique d'alarme avant le prétraitement. La figure 3.5 présente le premier résultat du prétraitement après avoir ciblé la chaudière 4 (cas d'étude). Les parties colorées en jaune sont les données de la chaudière 3, et les autres sont celles de la chaudière 4.

Date	Heur	Tag	Alarm
2022/02/01 08:57:04.390 +01:00	5:02:00	1051FSL5312_AL	1051FSL5312 FSL-5312-COMBUSTION AIR ALM
2022/02/01 08:57:05.040 +01:00	3:02:01	1051TI5409B	1051TI5409B SH2 TEMP. B PV = 985.6 DegC IOP
2022/02/01 08:57:05.040 +01:00	3:02:00	1051FI5313	1051FI5313 COMBUSTION AIR FLOW PV = 37906 Nm3/hr LL Recover
2022/02/01 08:57:06.090 +01:00	3:02:00	1051FI5313	1051FI5313 COMBUSTION AIR FLOW PV = 42338 Nm3/hr LO Recover
2022/02/01 08:57:06.090 +01:00	3:02:00	1051FI5313	1051FI5313 COMBUSTION AIR FLOW NR
2022/02/01 08:57:07.000 +01:00	3:00:00	1051FV5313B	1051FV5313B 2ND COMB. AIRFAN F.C.D MV = 5.0 % old= 0.0 [ONUSER@HIS0253]
2022/02/01 08:57:07.190 +01:00	3:02:00	1051PI5311	1051PI5311 COMBUSTION AIR PRESSURE PV = 66 mmH2O LO Recover
2022/02/01 08:57:07.190 +01:00	3:02:00	1051PI5311	1051PI5311 COMBUSTION AIR PRESSURE NR
2022/02/01 08:57:08.040 +01:00	3:02:01	1051TI5409B	1051TI5409B SH2 TEMP. B PV = 992.4 DegC IOP Recover
2022/02/01 08:57:22.040 +01:00	3:02:01	1051TI5409B	1051TI5409B SH2 TEMP. B PV = 1014.0 DegC IOP
2022/02/01 08:57:26.040 +01:00	3:02:01	1051TI5409B	1051TI5409B SH2 TEMP. B PV = 1007.1 DegC IOP Recover
2022/02/01 08:57:29.040 +01:00	3:02:01	1051TI5409B	1051TI5409B SH2 TEMP. B PV = 1004.1 DegC IOP
2022/02/01 08:57:29.000 +01:00	3:00:00	1051FV5313A	1051FV5313A 1ST COMB. AIRFAN F.C.D MV = 5.0 % old= 6.0 [ONUSER@HIS0253]
2022/02/01 08:57:31.040 +01:00	3:02:01	1051TI5409B	1051TI5409B SH2 TEMP. B PV = 1004.7 DegC IOP Recover
2022/02/01 08:57:34.040 +01:00	3:02:01	1051TI5409B	1051TI5409B SH2 TEMP. B PV = 1002.0 DegC IOP
2022/02/01 08:57:40.040 +01:00	3:02:01	1051TI5409B	1051TI5409B SH2 TEMP. B PV = 993.6 DegC IOP Recover
2022/02/01 08:57:43.040 +01:00	3:02:01	1051TI5409B	1051TI5409B SH2 TEMP. B PV = 997.9 DegC IOP
2022/02/01 08:57:44.000 +01:00	3:00:00	1051SS5378	1051SS5378 AUTO MANUAL SEL SWITCH PV =AUT old=MAN [ONUSER@HIS0253]
2022/02/01 08:57:47.040 +01:00	3:02:01	1051TI5409B	1051TI5409B SH2 TEMP. B PV = 1008.2 DegC IOP Recover
2022/02/01 08:57:50.040 +01:00	3:02:01	1051TI5409B	1051TI5409B SH2 TEMP. B PV = 1032.5 DegC IOP
2022/02/01 08:57:53.040 +01:00	3:02:01	1051TI5409B	1051TI5409B SH2 TEMP. B PV = 995.4 DegC IOP Recover

Figure 3.4: Historique avant sélection de la chaudière.

Date	Heure	Tag	Alarm
2022/02/01 00:00:06.340 +01:00	3:02:00	1051PI5318	1051PI5318 AIR BLOWER TO F.S. PRESS PV = 704.2 mmH2O LO Recover
2022/02/01 00:00:07.340 +01:00	3:02:00	1051PI5318	1051PI5318 AIR BLOWER TO F.S. PRESS NR
2022/02/01 00:00:37.040 +01:00	5:02:00	1051L4453AL	1051L4453AL INTERLOCK ALARM NR
2022/02/01 00:00:40.040 +01:00	5:02:00	1051PALL5318	1051PALL5318 AIR BLOWER TO F.S. PRESS NR
2022/02/01 00:01:27.290 +01:00	3:00:00	1051FV5313A	1051FV5313A 1ST COMB. AIRFAN F.C.D MV = 6.0 % old= 0.0 [ONUSER@HIS0253]
2022/02/01 00:01:29.290 +01:00	3:02:00	1051FI5312A	1051FI5312A COMBUSTION AIR FLOW Ch A PV = 38200 Nm3/hr LL Recover
2022/02/01 00:03:55.290 +01:00	3:02:00	1051FI5312B	1051FI5312B COMBUSTION AIR FLOW Ch B PV = 37436 Nm3/hr LL Recover
2022/02/01 00:03:58.290 +01:00	3:02:00	1051FI5312C	1051FI5312C COMBUSTION AIR FLOW Ch C PV = 37627 Nm3/hr LL Recover
2022/02/01 00:04:16.340 +01:00	3:02:00	1051FI5312	1051FI5312 COMBUSTION AIR AVE-FLOW PV = 37627 Nm3/hr LL Recover
2022/02/01 00:04:18.290 +01:00	3:02:00	1051FI5312A	1051FI5312A COMBUSTION AIR FLOW Ch A PV = 49469 Nm3/hr LO Recover
2022/02/01 00:04:59.290 +01:00	3:02:00	1051FI5312A	1051FI5312A COMBUSTION AIR FLOW Ch A NR
2022/02/01 00:05:01.290 +01:00	3:02:00	1051FI5312B	1051FI5312B COMBUSTION AIR FLOW Ch B PV = 47941 Nm3/hr LO Recover
2022/02/01 00:06:03.290 +01:00	3:02:00	1051FI5312B	1051FI5312B COMBUSTION AIR FLOW Ch B NR
2022/02/01 00:06:04.340 +01:00	3:02:00	1051FI5312C	1051FI5312C COMBUSTION AIR FLOW Ch C PV = 48705 Nm3/hr LO Recover
2022/02/01 00:06:26.340 +01:00	3:02:00	1051FI5312C	1051FI5312C COMBUSTION AIR FLOW Ch C NR
2022/02/01 00:06:28.340 +01:00	3:02:00	1051FI5312	1051FI5312 COMBUSTION AIR AVE-FLOW PV = 48705 Nm3/hr LO Recover
2022/02/01 00:06:44.040 +01:00	3:02:00	1051FI5312	1051FI5312 COMBUSTION AIR AVE-FLOW NR
2022/02/01 00:06:45.040 +01:00	5:02:00	1051FSL5312_AL	1051FSL5312 FSL-5312-COMBUSTION AIR ALM
2022/02/01 00:07:57.040 +01:00	3:02:00	1051FI5313	1051FI5313 COMBUSTION AIR FLOW PV = 37906 Nm3/hr LL Recover
2022/02/01 00:08:03.040 +01:00	3:02:00	1051FI5313	1051FI5313 COMBUSTION AIR FLOW PV = 42338 Nm3/hr LO Recover
2022/02/01 00:08:07.340 +01:00	3:02:00	1051FI5313	1051FI5313 COMBUSTION AIR FLOW NR
2022/02/01 00:08:09.290 +01:00	3:00:00	1051FV5313B	1051FV5313B 2ND COMB. AIRFAN F.C.D MV = 5.0 % old= 0.0 [ONUSER@HIS0253]

Figure 3.5: Historique après sélection de la chaudière.

Notre fichier d'historique d'alarme comportait aussi des alarmes nuisibles à notre algorithme d'apprentissage qu'on a dû supprimer grâce au prétraitement. Le tableau 3.1 et le tableau 3.2 montrent les alarmes systèmes ainsi que les modes opératoires présents dans nos données d'historique d'alarme :

Alarmes systèmes		
RECOVER ALARM SIGNAL AFTER A LOSS [RECOVER]	1051LI5303C	STEAM DRUM LEVEL Ch C PV = 38.8 % LO Recover
ALARM THAT OCCURE ONLY TO THIS USER [ONUSER]	1051LIC5304B	STEAM DRUM LEVEL MV = 14.5 % old= 9.0 [ONUSER@HIS0252]

Tableau 3.1 : Exemples des alarmes systèmes.

Modes operatories		
Manual [MAN]	1051LIC5304B	STEAM DRUM LEVEL MAN
AUTOMATIQUE [AUT]	1051LIC5304B	STEAM DRUM LEVEL AUT

MANUAL INITIALIZATION [IMAN]	1051FIC5321	FG FLOW TO BURNERS CAS IMAN
FOLLOWING EXIT [TRK]	1051FV5313B	2ND COMB. AIRFAN F.C.D AUT TRK
CASCADE [CAS]	1051FIC5321	FG FLOW TO BURNERS CAS IMAN

Tableau 3.2 : Exemples des modes opératoires

La figure 3.6 suivante montre le journal d'historique d'alarme avant la suppression des alarmes nuisibles, on peut constater que nos données d'historiques d'alarmes contiennent beaucoup de données nuisibles à notre algorithme d'apprentissage qu'on doit supprimer (les parties colorées en jaune sont les alarmes systèmes, et les parties colorées en bleu sont les modes opératoires). Et la figure 3.7 présente le résultat de la suppression des alarmes nuisibles grâce au prétraitement.

Date	Heure	Tag	Alarm
2022/02/01 09:07:42.040 +01:00	3:02:00	1051AI5391	1051AI5391 BFW TOTAL CONDUCTIVITY PV = 5.1 us/cm HI
2022/02/01 09:08:02.040 +01:00	3:02:00	1051AI5392	1051AI5392 BFW PH PV = 8.75 pH LO Recover
2022/02/01 09:08:02.040 +01:00	3:02:00	1051AI5392	1051AI5392 BFW PH NR
2022/02/01 09:36:45.000 +01:00	4:00:00	1051TIC5310	1051TIC5310 BFW TEMP CTRL TO DSH MV = 1.8 % old= 0.0 [ONUSER@HIS0251]
2022/02/01 09:46:40.090 +01:00	3:02:00	1051TI5302	1051TI5302 BFW TEMP. ECON. INLET PV = 119.0 DegC LO Recover
2022/02/01 09:46:40.090 +01:00	3:02:00	1051TI5302	1051TI5302 BFW TEMP. ECON. INLET NR
2022/02/01 10:11:43.000 +01:00	4:00:00	1051TIC5310	1051TIC5310 BFW TEMP CTRL TO DSH MV = 4.8 % old= 1.8 [ONUSER@HIS0253]
2022/02/01 10:16:37.000 +01:00	4:00:00	1051TIC5310	1051TIC5310 BFW TEMP CTRL TO DSH MV = 9.1 % old= 4.8 [ONUSER@HIS0253]
2022/02/01 10:41:31.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B PV = 505.5 DegC HI
2022/02/01 10:41:31.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B PV = 505.5 DegC HH
2022/02/01 10:41:32.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B PV = 368.0 DegC HH Recover
2022/02/01 10:41:32.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B PV = 368.0 DegC HI Recover
2022/02/01 10:41:32.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B NR
2022/02/01 10:43:45.000 +01:00	4:00:00	1051TIC5310	1051TIC5310 BFW TEMP CTRL TO DSH MV = 11.0 % old= 9.1 [ONUSER@HIS0252]
2022/02/01 10:44:19.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B PV = 225.6 DegC IOP
2022/02/01 10:44:20.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B PV = 229.1 DegC IOP Recover
2022/02/01 10:44:20.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B NR
2022/02/01 10:44:24.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B PV = 227.4 DegC IOP
2022/02/01 10:44:27.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B PV = 331.8 DegC IOP Recover
2022/02/01 10:44:27.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B NR
2022/02/01 10:44:29.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B PV = 331.8 DegC IOP
2022/02/01 10:44:31.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B PV = 226.6 DegC IOP Recover
2022/02/01 10:44:31.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B NR
2022/02/01 10:44:55.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B PV = 225.5 DegC IOP
2022/02/01 10:44:56.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B PV = 940.7 DegC IOP Recover
2022/02/01 10:44:56.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B PV = 940.7 DegC HI
2022/02/01 10:44:56.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B PV = 940.7 DegC HH
2022/02/01 10:44:59.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B PV = 986.2 DegC IOP
2022/02/01 10:45:00.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B PV = 280.3 DegC IOP Recover
2022/02/01 10:45:00.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B PV = 280.3 DegC HH Recover
2022/02/01 10:45:00.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B PV = 280.3 DegC HI Recover
2022/02/01 10:45:00.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B NR
2022/02/01 10:45:18.000 +01:00	4:00:00	1051TIC5310	1051TIC5310 BFW TEMP CTRL TO DSH MV = 14.2 % old= 11.0 [ONUSER@HIS0252]
2022/02/01 10:45:29.000 +01:00	4:00:00	1051TIC5310	1051TIC5310 BFW TEMP CTRL TO DSH MV = 17.3 % old= 14.2 [ONUSER@HIS0252]

Figure 3.6: Avant suppression des alarmes parasites.

Date	Heure	Tag	Alarm
2022/02/01 09:07:42.040 +01:00	3:02:00	1051AI5391	1051AI5391 BFW TOTAL CONDUCTIVITY PV = 5.1 us/cm HI
2022/02/01 09:08:02.040 +01:00	3:02:00	1051AI5392	1051AI5392 BFW PH NR
2022/02/01 09:46:40.090 +01:00	3:02:00	1051TI5302	1051TI5302 BFW TEMP. ECON. INLET NR
2022/02/01 10:41:31.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B PV = 505.5 DegC HI
2022/02/01 10:41:31.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B PV = 505.5 DegC HH
2022/02/01 10:41:32.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B NR
2022/02/01 10:44:19.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B PV = 225.6 DegC IOP
2022/02/01 10:44:20.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B NR
2022/02/01 10:44:24.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B PV = 227.4 DegC IOP
2022/02/01 10:44:27.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B NR
2022/02/01 10:44:29.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B PV = 331.8 DegC IOP
2022/02/01 10:44:31.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B NR
2022/02/01 10:44:55.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B PV = 225.5 DegC IOP
2022/02/01 10:44:56.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B PV = 940.7 DegC HI
2022/02/01 10:44:56.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B PV = 940.7 DegC HH
2022/02/01 10:44:59.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B PV = 986.2 DegC IOP
2022/02/01 10:45:00.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B NR
2022/02/01 10:46:20.040 +01:00	3:02:01	1051TI5309B	1051TI5309B SH2 TEMP. B PV = 224.5 DegC IOP

Figure 3.6: Après suppression des alarmes parasites.

Nos données d'historique d'alarme contiennent aussi des alarmes redondantes qui s'activent consécutivement sans avoir un retour, qu'on doit supprimer à l'aide du prétraitement. La figure 3.8 suivante montre des alarmes redondantes dans notre historique d'alarme.

	Date	Tag	Alarm	Fault
26	2022/02/01 00:22:55.040 +01:00	1051PI5324A	62 1051PI5324A F.GAS PRESS.TO BRNS.Ch.A PV = 2.130 kg/cm2 HI	
27	2022/02/01 00:22:58.040 +01:00	1051PI5324A	62 1051PI5324A F.GAS PRESS.TO BRNS.Ch.A PV = 2.130 kg/cm2 HH	
28	2022/02/01 00:24:26.340 +01:00	1051PI5324B	63 1051PI5324B F.GAS PRESS.TO BRNS.Ch.B PV = 2.140 kg/cm2 HI	
29	2022/02/01 00:24:28.340 +01:00	1051PI5324B	63 1051PI5324B F.GAS PRESS.TO BRNS.Ch.B PV = 2.140 kg/cm2 HH	
30	2022/02/01 00:25:05.340 +01:00	1051PI5324C	64 1051PI5324C F.GAS PRESS.TO BRNS.Ch.C PV = 2.075 kg/cm2 HI	
31	2022/02/01 00:25:07.340 +01:00	1051PI5324C	64 1051PI5324C F.GAS PRESS.TO BRNS.Ch.C PV = 2.075 kg/cm2 HH	
32	2022/02/01 00:25:43.340 +01:00	1051PI5324	61 1051PI5324 FUEL GAS PRESS BRNS. AVE PV = 2.115 kg/cm2 HI	
33	2022/02/01 00:25:45.340 +01:00	1051PI5324	61 1051PI5324 FUEL GAS PRESS BRNS. AVE PV = 2.115 kg/cm2 HH	
34	2022/02/01 00:25:48.340 +01:00	1051PIC5324	65 1051PIC5324 FG FLOW TO BURNERS PV = 2.115 kg/cm2 HI	
35	2022/02/01 00:25:50.390 +01:00	1051PIC5324	65 1051PIC5324 FG FLOW TO BURNERS PV = 2.115 kg/cm2 HH	
36	2022/02/01 00:26:05.340 +01:00	1051FI5321A	11 1051FI5321A FG FLOW TO BURNERS PV = 911 kg/hr LO	
37	2022/02/01 00:26:06.340 +01:00	1051FI5321	10 1051FI5321 FG M.H PRESS(UPSTREAM) PV = 1537.5 Nm3/hr LO	
38	2022/02/01 00:26:59.990 +01:00	1051PSHH5324_AL	-66 1051PSHH5324 PSHH5324-FUEL GAS BURNER NR	

Figure 3.8: Alarmes redondantes

Pour le bon fonctionnement de notre apprentissage on doit codifier les Tags (instrument qui déclenche l'alarme), et donner à chaque Tag un entier unique.

	Date	Tag	Alarm	NumTag
0	2022/02/01 00:00:06.340 +01:00	1051PI5318	1051PI5318 AIR BLOWER TO F.S. PRESS PV = 704.2 mmH2O LO Recover	59
1	2022/02/01 00:00:07.340 +01:00	1051PI5318	1051PI5318 AIR BLOWER TO F.S. PRESS NR	59
2	2022/02/01 00:00:37.040 +01:00	1051IL4453AL	1051IL4453AL INTERLOCK ALARM NR	42
3	2022/02/01 00:00:40.040 +01:00	1051PALL5318	1051PALL5318 AIR BLOWER TO F.S. PRESS NR	56
4	2022/02/01 00:01:27.290 +01:00	1051FV5313A	1051FV5313A 1ST COMB. AIRFAN F.C.D MV = 6.0 % old= 0.0 [ONUSER@HIS0253]	15
5	2022/02/01 00:01:29.290 +01:00	1051FI5312A	1051FI5312A COMBUSTION AIR FLOW Ch A PV = 38200 Nm3/hr LL Recover	6
6	2022/02/01 00:03:55.290 +01:00	1051FI5312B	1051FI5312B COMBUSTION AIR FLOW Ch B PV = 37436 Nm3/hr LL Recover	7
7	2022/02/01 00:03:58.290 +01:00	1051FI5312C	1051FI5312C COMBUSTION AIR FLOW Ch C PV = 37627 Nm3/hr LL Recover	8
8	2022/02/01 00:04:16.340 +01:00	1051FI5312	1051FI5312 COMBUSTION AIR AVE-FLOW PV = 37627 Nm3/hr LL Recover	5
9	2022/02/01 00:04:18.290 +01:00	1051FI5312A	1051FI5312A COMBUSTION AIR FLOW Ch A PV = 49469 Nm3/hr LO Recover	6
10	2022/02/01 00:04:59.290 +01:00	1051FI5312A	1051FI5312A COMBUSTION AIR FLOW Ch A NR	6

Figure 3.9: Codification des Tags

Nos données d'historique d'alarme contiennent des déclenchements d'alarme et des retours d'alarmes qu'on doit identifier. La figure 3.10 suivante montre les déclenchements d'alarme ALM et les retours d'alarme NR.

	Date	Tag	Alarm	Fault
0	2022/02/01 00:00:07.340 +01:00	1051PI5318	-59	1051PI5318 AIR BLOWER TO F.S. PRESS NR
1	2022/02/01 00:00:37.040 +01:00	1051IL4453AL	-42	1051IL4453AL INTERLOCK ALARM NR
2	2022/02/01 00:00:40.040 +01:00	1051PALL5318	-56	1051PALL5318 AIR BLOWER TO F.S. PRESS NR
3	2022/02/01 00:04:59.290 +01:00	1051FI5312A	-6	1051FI5312A COMBUSTION AIR FLOW Ch A NR
4	2022/02/01 00:06:03.290 +01:00	1051FI5312B	-7	1051FI5312B COMBUSTION AIR FLOW Ch B NR
5	2022/02/01 00:06:26.340 +01:00	1051FI5312C	-8	1051FI5312C COMBUSTION AIR FLOW Ch C NR
6	2022/02/01 00:06:44.040 +01:00	1051FI5312	-5	1051FI5312 COMBUSTION AIR AVE-FLOW NR
7	2022/02/01 00:06:45.040 +01:00	1051FSL5312_AL	14	1051FSL5312 FSL-5312-COMBUSTION AIR ALM
8	2022/02/01 00:08:07.340 +01:00	1051FI5313	-9	1051FI5313 COMBUSTION AIR FLOW NR
9	2022/02/01 00:11:13.040 +01:00	1051PI5311	-58	1051PI5311 COMBUSTION AIR PRESSURE NR
10	2022/02/01 00:11:56.340 +01:00	1051FALL5312_AL	-4	1051FALL5312 COMBUSTION AIR FLL LOW NR

Figure 3.10: Etat de l'alarme

Dans la figure 3.11 suivante on peut constater que chaque évènement est associé à un temps d'occurrence donné sous la forme d'un nombre entier.

6. Validation

La validation est le processus d'évaluation de la performance des modèles Data Mining par rapport aux données réelles. Il est important de valider les modèles Data Mining en comprenant leurs qualités et leurs caractéristiques avant de les déployer dans un environnement de production. Il existe plusieurs méthodes de test et de validation des modèles de Data Mining dont :

- Utiliser diverses mesures de validité statistique pour déterminer s'il existe des problèmes dans les données ou dans le modèle ;
- Séparer les données en deux données d'apprentissage, et des données pour vérifier l'exactitude des prédictions ;
- Demander à des experts du domaine d'examiner les résultats du modèle Data Mining pour déterminer si les modèles découverts ont une signification dans le scénario du domaine ciblé ;

Toutes ces méthodes sont utiles dans la méthodologie Data Mining et sont utilisées de manière itérative lorsque vous créez, testez et affinez des modèles pour répondre à un problème spécifique. Aucune règle globale unique ne peut dire qu'un modèle est suffisamment bon. Dans notre validation on va utiliser l'avis d'un expert (un opérateur DCS) pour examiner nos résultats du modèle Data Mining.

Pour évaluer pleinement l'efficacité d'un modèle, on doit examiner à la fois la précision et le rappel. La précision et le rappel sont souvent liés l'un à l'autre, l'amélioration de la précision réduit généralement le rappel et vice versa.

6.1. Précision

La précision est une mesure de la façon dont le modèle corrèle un résultat avec les attributs des données fournies. Il existe plusieurs mesures de la précision, mais toutes dépendent des données utilisées. En réalité, les valeurs peuvent être manquantes ou approximatives, ou les données peuvent avoir été modifiées par de multiples processus. En particulier dans la phase d'exploration et de développement, On peut décider d'accepter une certaine quantité d'erreurs dans les données, surtout si les données sont assez uniformes dans leurs caractéristiques.

Définition mathématique de la précision [33].

$$Précision = \frac{Tp}{Tp + Fp}$$

6.2.Rappel

Le rappel c'est le ratio du nombre d'éléments pertinents retrouvés par le nombre total d'éléments pertinents.

Définition mathématique du rappel [33].

$$Rappel = \frac{Tp}{Tp + Fn}$$

Tp	Le taux de vrais positifs, c'est-à-dire le nombre d'instances qui sont pertinentes et que le modèle a correctement identifiées comme pertinentes.
Fp	Le taux de faux positifs, c'est-à-dire le nombre d'instances qui ne sont pas pertinentes mais que le modèle a incorrectement identifiées comme pertinentes.
Fn	Le taux de faux négatifs, c'est-à-dire le nombre d'instances qui sont pertinentes et que le modèle a incorrectement identifiées comme non pertinentes.

Tableau 3.3: Explication des symboles des formules Precision et Recall

6.3.Calcul de la précision et du rappel

Pour calculer la précision et le rappel, on a d'abord trié nos résultats d'apprentissage en 4 parties, faux positif, faux négatif, vrai positif et vrai négatif grâce à notre expert du domaine (l'opérateur DCS).

Vrai Positif	65
Vrai Positif	4
Faux Positif	7
Faux négatif	18
	94

Tableau 3.4 : Résultats de notre algorithme

Calcul de la Précision de notre Modèle :

$$Précision = \frac{Tp}{Tp + Fp} = \frac{65}{65 + 7} = 0.90$$

Calcul du Rappel de notre Modèle

$$Rappel = \frac{Tp}{Tp + Fn} = \frac{65}{65 + 18} = 0.78$$

Pour décrire les performances d'un modèle, la valeur la plus importante est le F-mesure, qui est la moyenne harmonique de la précision et du rappel :

Définition mathématique du F-mesure [33].

$$F = 2 * \frac{Precision \times Recall}{Precision + Recall} = 2 * \frac{0.9 * 0.78}{0.9 + 0.78} = 0.83$$

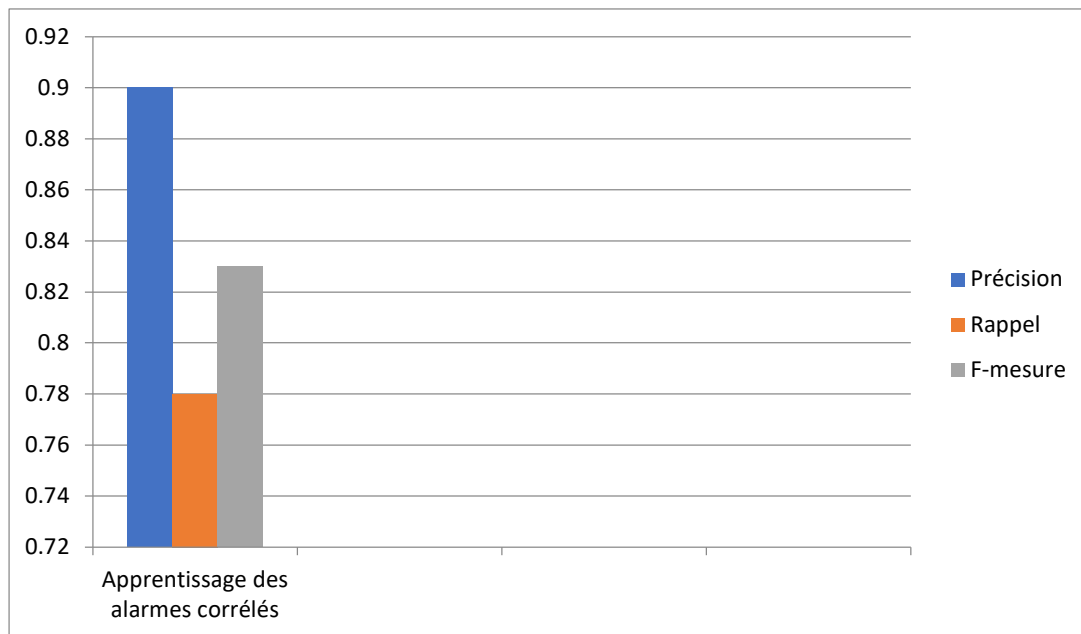


Figure 3.15: Résultats de précision et de rappel du modèle d'apprentissage.

6.4.Discussion

De nombreux travaux ont été réalisés pour analyser et prédire les corrélations des alarmes industrielles. Pour la validation nous nous sommes concentrés sur les mesures (précision, rappel, et F-score) qui nous permettent d'évaluer notre modèle Data Mining.

Nous avons obtenu de bons résultats de précision et de rappel malgré la complexité des données d'historique d'alarmes réel. Cette complexité était la cause des faux négatif (FN) estimés de 19% des résultats. Les faux négatifs sont dus aussi aux alarmes déclenchées mais qui n'ont pas eu de retour, c'est pour ça que ces données n'ont pas pu avoir une corrélation. Il y'a eu aussi des corrélations avec un pourcentage faible qui n'ont pas été approuvé par notre expert 4%. Ces résultats sont considérés comme vrai négatif (TN).

7. Conclusion

Dans ce chapitre nous avons présenté une architecture générale d'un système d'apprentissage pour l'extraction efficace et significative de modèles d'alarmes à partir d'une séquence d'événements d'alarmes pour la surveillance et la supervision industrielle. Certains composants de cette architecture ont été implémentés pour illustrer non seulement son fonctionnement mais aussi les possibilités offertes pour l'opérateur mais aussi pour la sécurité. Cette implémentation de ce système sera présentée dans le prochain chapitre.

Implémentation

1. Introduction

Après avoir effectué l'étude conceptuelle de notre système, nous passons à l'implémentation de l'application définie et détaillée au chapitre précédent où on va le convertir en un préliminaire plus compréhensible par la machine. On va présenter tout d'abord l'ensemble des outils utilisés (plateforme, éditeurs...) ainsi que différents langages pour atteindre notre produit final, ensuite nous illustrons des captures d'écrans de notre logiciel.

2. Les outils utilisés dans le développement



Figure 4.1: Les outils utilisés dans le développement

1.1. Visual Studio Code

Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS.[34] Il est livré avec une prise en charge intégrée de JavaScript,

TypeScript et Node.js et dispose d'un riche écosystème d'extensions pour d'autres langages (tels que C++, C#, Java, Python, PHP, Go) et moteurs d'exécution (tels que .NET et Unity) [35].

1.2.Nodejs

Nodejs est un environnement d'exécution javascript open source, multi plateformes utilisé pour donner aux applications javascript la possibilité pour s'exécuter sur le serveur, pour cela Nodejs prend l'avantage du moteur d'exécution Javascript V8 (V8 JavaScript runtime engine) Le moteur V8 est écrit en C++ et utilisé dans Chrome et avec Node.js, V8 joue le rôle d'un pont qui prend le code et le convertit à un code machine plus rapide qui peut être exécuté nativement sans interprétation [36].

1.3.React

React est une bibliothèque JavaScript libre développée par Facebook depuis 2013 qui peut être utilisé sur les plates-formes mobiles et de bureau avec de nombreuses fonctionnalités distinctes. L'une d'elles est la capacité de créer des pages interactives plutôt que de simples pages statiques qui peuvent mettre à jour et rendre les données après chaque entrée de l'utilisateur, ce qui permet une interface utilisateur transparente sans avoir à rafraîchir la page entière à chaque fois qu'un changement est effectué. La raison est que la structure de React est basée sur des composants qui permettent de concevoir des interfaces utilisateur complexes, la logique des composants étant écrite en JavaScript [37].

1.4.Express

Express est une infrastructure d'applications Web Node.js open source et gratuit, minimaliste et flexible qui fournit un ensemble de fonctionnalités robuste pour les applications Web et mobiles [38].

1.5.ESLint

ESLint est un utilitaire de linting JavaScript open source créé à l'origine par Nicholas C. Zakas en juin 2013. Le linting de code est un type d'analyse statique qui est fréquemment utilisé pour trouver des modèles problématiques ou du code qui n'adhère pas à certaines directives de style. Il existe des linters de code pour la plupart des langages de programmation, et les compilateurs intègrent parfois le linting dans le processus de compilation. JavaScript, étant un langage dynamique et faiblement typé, est particulièrement sujet aux erreurs des développeurs. Sans l'avantage d'un processus de compilation, le code JavaScript est généralement exécuté afin

de trouver des erreurs de syntaxe ou autres. Les outils de linting comme ESLint permettent aux développeurs de découvrir les problèmes de leur code JavaScript sans l'exécuter [39].

1.6.Anaconda

Anaconda est la plateforme de science des données la plus populaire au monde et la base de l'apprentissage automatique moderne. Anaconda est un logiciel open-source [40] des langages de programmation Python et R pour le développement d'applications dédiées à la science des données et à l'apprentissage automatique (traitement de données à grande échelle, analyse prédictive, calcul scientifique), conçue pour simplifier la gestion et le déploiement des packages [41]. Les versions de package sont gérées par le système de gestion de packages conda [42].

1.7.Python

Python est un langage de programmation interprété, interactif et orienté objet. Il intègre des modules, des exceptions, un typage dynamique, des types de données dynamiques de très haut niveau et des classes. Il prend en charge plusieurs paradigmes de programmation au-delà de la programmation orientée objet, tels que la programmation procédurale et fonctionnelle. Python combine une puissance remarquable avec une syntaxe très claire. Il possède des interfaces avec de nombreux appels système et bibliothèques, ainsi qu'avec divers systèmes de fenêtres, et est extensible en C ou C++. Il est également utilisable comme langage d'extension pour les applications qui ont besoin d'une interface programmable. Enfin, Python est portable : il fonctionne sur de nombreuses variantes d'Unix, dont Linux et macOS, et sur Windows [43].

1.8.Les bibliothèques utilisées

1.8.1. Numpy est le package fondamental pour le calcul scientifique en Python. Il s'agit d'une bibliothèque Python qui fournit un objet tableau multidimensionnel, divers objets dérivés (tels que des tableaux et des matrices masqués) et un assortiment de routines pour des opérations rapides sur des tableaux, notamment mathématiques, logiques, manipulation de forme, tri, sélection, E/S, transformées de Fourier discrètes, algèbre linéaire de base, opérations statistiques de base, simulation aléatoire et bien plus encore. [44]

1.8.2. Pandas est une bibliothèque open source sous licence BSD fournissant des structures de données et des outils d'analyse de données hautes performances et faciles à utiliser pour le langage de programmation Python [45].

- 1.8.3. Matplotlib** est une bibliothèque complète pour créer des visualisations statiques, animées et interactives en Python, il offre une alternative open source viable à MATLAB [47].
- 1.8.4. Json** en Python est un format standard inspiré de JavaScript pour l'échange et le transfert de données au format texte sur un réseau. Généralement, JSON est au format chaîne ou texte. Il peut être utilisé par les API et les bases de données, et il représente les objets sous forme de paires nom/valeur. JSON est l'abréviation de JavaScript Object Notation [46].
- 1.8.5. Flask** est un petit framework web écrit en Python. Il est classé comme un micro-framework car il ne nécessite aucun outil ou bibliothèque spécifique [48]. Il n'a pas de couche d'abstraction de base de données, de validation de formulaire ou d'autres composants de bibliothèque tiers préexistants pour lesquels il fournit des fonctionnalités communes. Cependant, Flask prend en charge les extensions qui peuvent ajouter des fonctionnalités à votre application, tout comme elles sont implémentées dans Flask lui-même. Il existe des mappers relationnels objet, la validation de formulaires, la gestion des téléchargements, diverses techniques d'authentification ouvertes et des extensions de plusieurs outils liés au cadre commun.
- 1.8.6. Seaborn** est une bibliothèque permettant de réaliser des graphiques statistiques en Python. Elle s'appuie sur matplotlib et s'intègre étroitement aux structures de données pandas. Seaborn aide à explorer et à comprendre les données. Ses fonctions de traçage opèrent sur des dataframes et des tableaux contenant des ensembles de données entiers et effectuent en interne la mise en correspondance sémantique et l'agrégation statistique nécessaires pour produire des tracés informatifs. Son API déclarative, orientée vers les ensembles de données, permet de se concentrer sur la signification des différents éléments des graphiques, plutôt que sur les détails de leur dessin [49].

2. Les étapes de création de service web

2.1. Backend (Flask)

Pour créer le service Web, nous avons réalisé les étapes suivantes :

Comme première étape de développement de service web, on a créé et activé l'environnement python.

Etape 1 : Création et l'activation d'un nouvel conda environnement

Nous créons un nouvel environnement python pour spécifier la version de python et isoler les paquets que nous devons installer pour ce travail.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS G:\Github\todo\APP> conda create --name AlarmService
```

Figure 4.2: Création d'un nouvel environnement Conda nommé AlarmService

```
PS G:\Github\todo\APP> conda activate AlarmService
PS G:\Github\todo\APP>
```

Figure 4.3: activation de l'environnement AlarmService

Etape 2 : Création de back-end classes

Dans cette étape, nous créons les classes back-end <leaningAlgorithm.py> <preprocessing_Data.py> <Relaion_Discovry.py> et la classe principale <app.py>

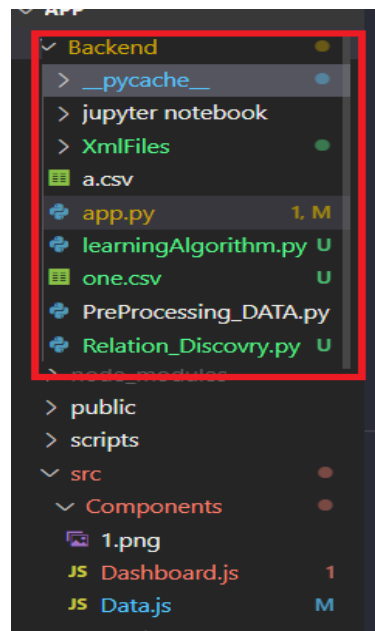


Figure 4.4: Classes et fichiers back-end

Etape 3 : Code source snap

Nous montrons dans cette figure 4.5 un extrait du code du fichier principal app.py

```
1  from flask import Flask,render_template
2  from flask import jsonify
3  import json
4  import pandas as pd
5  import matplotlib.pyplot as plt
6  from matplotlib import pyplot
7  import PreProcessing_DATA as PreP_D
8  import Relation_Discovry as Rule_Discovry
9  import numpy as np
10 import seaborn as sns
11 from json import loads
12 from dicttoxml import dicttoxml
13 import xml.etree.ElementTree as ET
14
15
16
17
18
19
20 app = Flask(__name__)
21
22 df = pd.read_csv('one.csv')
23 @app.route('/table')
24 def table():
25     data = pd.read_csv('one.csv')
26
27     return data.to_json(orient='records')
28
29 @app.route('/PreProData')
30 def PreProData():
31     data = pd.read_csv('a.csv')
32     DFAlarms = data.drop(['Heure'],axis=1)
33     DfAlarm=PreP_D.Preprocecssing.NumTag(DFAlarms)
34     DfAlarm=PreP_D.Preprocecssing.checkNumTag(DFAlarms)
35     DfAlarm=PreP_D.Preprocecssing.AlarmState(DFAlarms)
36     DfAlarm.dropna(inplace=True)
37     DfAlarm =DfAlarm.reset_index(drop=True)
38     DfAlarm=PreP_D.Preprocecssing.SequenceAlarm(DfAlarm)
39     DfAlarm =DfAlarm.reset_index(drop=True)
40     DfAlarm=PreP_D.Preprocecssing.AlarmTime(DfAlarm)
41     #df.to_json(r'D:\File Name.json', orient='index')
42     json_obj =DfAlarm.to_json(orient='records')
43     xml = dicttoxml(loads(json_obj))
44     with open("XmlFiles/Preprocessing_DATA.xml", "wb") as f:
45         f.write(xml)
46     return json_obj
```

Figure 4.5: Code source de la classe « app.py »

Étape 4 : Tester le back-end

Dans cette étape, nous testons le back-end en utilisant "POSTMAN" pour tester le post et le get des API de notre serveur.

```
PROBLEMS 2 OUTPUT TERMINAL DEBUG CONSOLE AZURE JUPYTER

* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with watchdog (windowsapi)
* Debugger is active!
* Debugger PIN: 103-774-823
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Figure 4.6: Démarrage du serveur

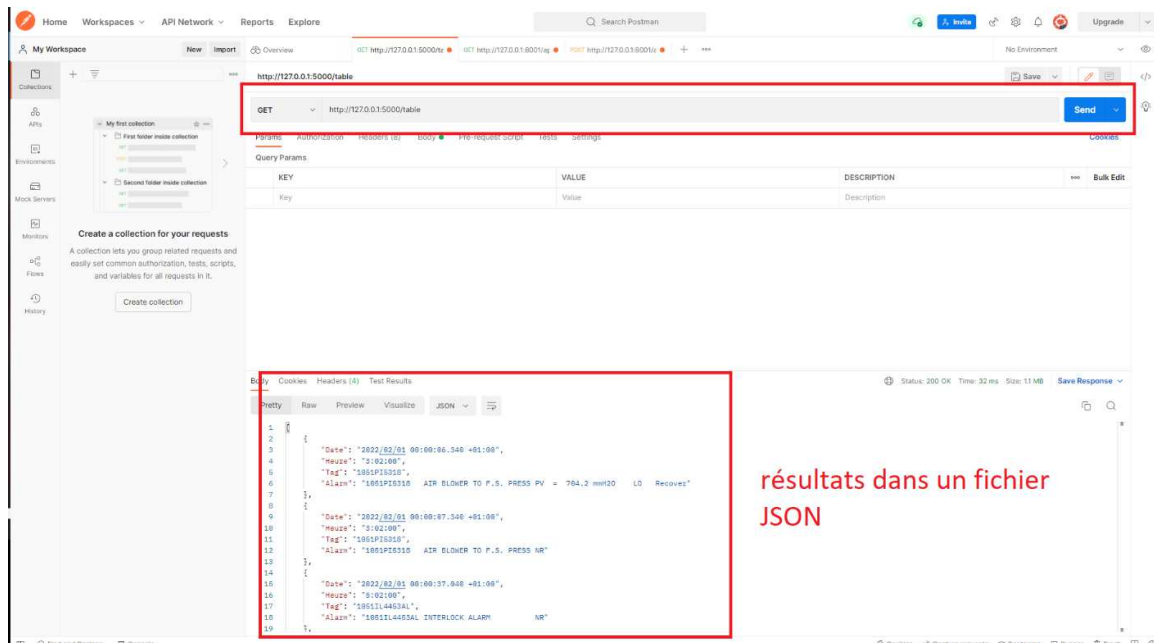


Figure 4.7: PostMan : Résultat de test back-end preprocessing

2.2. Front-End (ReactJS & ElectronJS)

Etape 1: Configuration ReactJs avec ElectronJs

Dans cette etape nous allons Exécuter les commandes sur le terminal du VScode pour installer des dépendances de reactJS et electronJS

```
$npx create-react-app AlarmService  
$cd AlarmService  
$Yarn install  
$Yarn Add electron, electron-is-dev, concurrently, wait-on
```

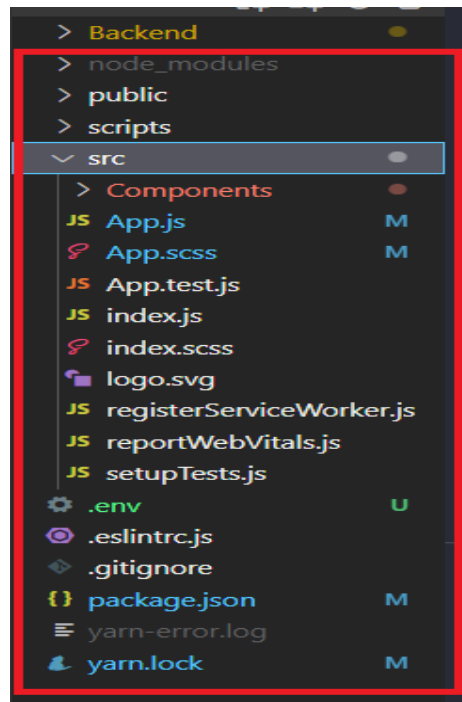


Figure 4.8: Front-end files

Etape 2 : Code source snap

La figure 4.9 montre un extrait du code de la classe principale du front-end app.js, c'est la fenêtre qui relie tous les composants, et la figure 4.10 est un extrait du code du composant preprocessing.js.

```
1 import React,{useState} from 'react';
2 import { MemoryRouter as Router, Routes, Route } from 'react-router-dom';
3 import './App.scss';
4
5 import { Row,Col} from 'react-bootstrap';
6 import 'bootstrap/dist/css/bootstrap.min.css';
7 import SideBare from './Components/SideBare';
8 import PreProData from './Components/PreProData';
9 import Data from './Components/Data';
10 import Dashboard from './Components/Dashboard';
11
12
13 const Apps = () => {
14   const target = {
15     Dashboard:"Dashboard",
16     PrePro: "PrePro",
17     data: "data",
18   };
19   const [displayContent, setDisplayContent] = useState(target.Dashboard);
20
21   return (
22     <Row >
23     <Col Col xs={6} md={4} >
24       <SideBare target={target} setDisplayContent={setDisplayContent} />
25     </Col>
26     <Col xs={12} md={8}>
27       {displayContent ===target.PrePro ?(<PreProData />
28       ): displayContent === target.data ?( <Data />
29       ): displayContent === target.Dashboard ?( <Dashboard /> ):
30
31       ( console.log('nothing sorry'))
32     </Col>
33     </Row>
34
35   );
36 };
```

Figure 4.9: App.js code source

```
1  import React, {useState, useEffect} from 'react';
2  import { Table } from 'react-bootstrap';
3  import 'bootstrap/dist/css/bootstrap.min.css';
4
5  const PreProData = () => {
6    const [data, setData] = useState([]);
7    useEffect(() => {
8      fetch("http://localhost:5000/PreProData")
9        .then(res => res.json())
10       .then(data => {
11         setData(data);
12         console.log(data)
13       });
14
15   }, [])
16   return (
17     <>
18
19     <h1>PreProcecing Data Table</h1>
20     <div className='PreProData'>
21       <div className="container">
22         <Table striped bordered hover size="sm" >
23           <thead>
24             <tr>
25               <th>#</th>
26               <th>Date</th>
27               <th>Tag </th>
28               <th>Time</th>
29               <th>Alarm</th>
30             </tr>
31           </thead>
32           <tbody>
33             {data.map((data, i) => (
34               <tr key={i}>
35                 <td>{i}</td>
36                 <td>{data.Date}</td>
37                 <td>{data.Tag}</td>
38                 <td>{data.Time}</td>
39                 <td>{data.Alarm}</td>
40               </tr>
41             ))}
42           </tbody>
43         </Table>
44       </div>
45     </div>
46   </>
47 );
48 };
49 export default PreProData ;
```

Figure 4.10 : Preprocessing window Code source

Etape 3 : Configuration du backend et frontend

Dans cette étape nous vous montrons le script qui exécute le back-end et le front-end simultanément.

```

1  "main": "public/main.js",
2  "homepage": "./",
3  "scripts": {
4    "start": "react-scripts start",
5    "build": "react-scripts build",
6    "test": "react-scripts test --env=jsdom",
7    "eject": "react-scripts eject",
8    "startsrv": "python app.py --prefix backend",
9    "electron:serve": "concurrently -k \"cross-env BROWSER=none yarn start\" \"yarn electron:start\"",
10   "electron:start": "wait-on tcp:3000 & electron ."
11  },
12  "eslintConfig": {
13    "extends": [
14      "react-app",
15      "react-app/jest"
16    ]
17  },

```

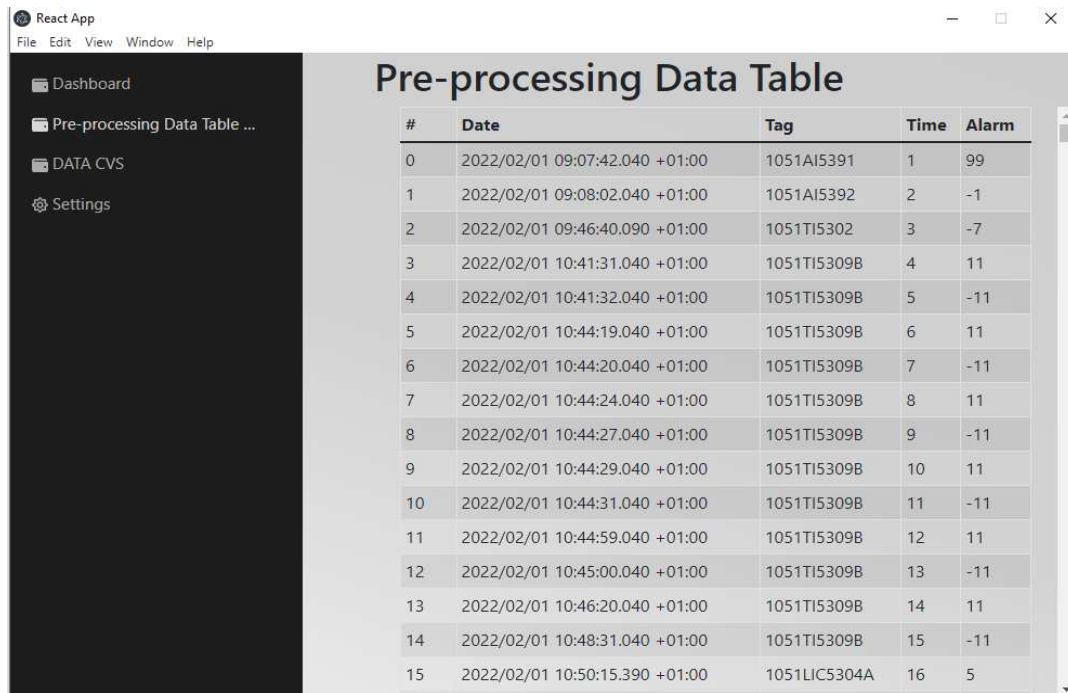
Figure 4.11: « package.js » configurant backend/frontend

2.3. Quelques interfaces de service web

Dans cette partie nous allons visualiser quelques aspects relatifs à l'interface du service Web.

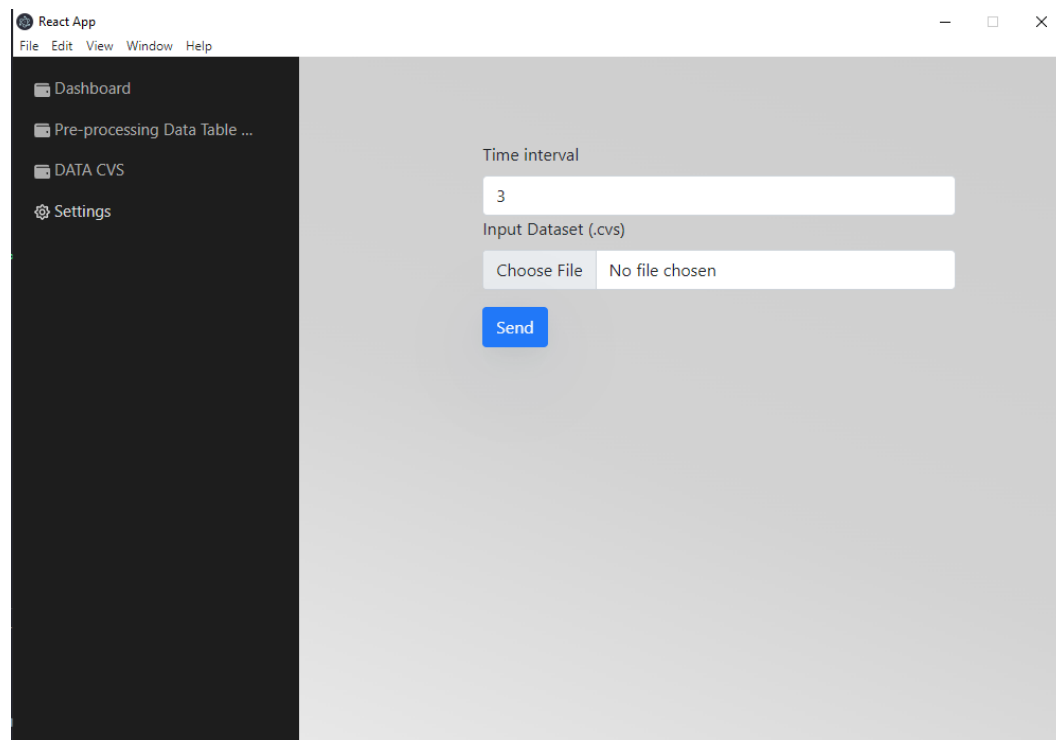
ALARM	Tag	Simple_Sliding %	User_Sliding %	Proposed
1051AI5392 BFW PH PV = 8.50 pH LO	1	1 => [5, 6, 4, 3] 33.33	1 => [7, 5, 6] 33.33	1 => [7, 5, 6] 33.33
1051LI5303A STEAM DRUM LEVEL Ch A PV = 34.6 % LO	2	2 => [4, 3] 5.59	2 => [4, 3] 5.59	2 => [4, 3] 5.59
1051LI5303B STEAM DRUM LEVEL Ch B PV = 34.7 % LO	3	3 => [2, 4, 2] 0.28	3 => [2, 5, 6, 4] 0.28	3 => [2, 5, 6, 4] 0.28
1051LI5303C STEAM DRUM LEVEL Ch C PV = 34.7 % LO	4	4 => [3, 2] 82.0	4 => [2, 3, 5, 6] 0.29	4 => [3, 2] 82.0
1051LIC5304A STEAM DRUM LEVEL PV = 40.0 % LO	5	5 => [6, 8, 10, 12, 9, 8, 10, 9, 4, 3, 2, 4, 3, 2, 4, 3, 2, 4, 3, 2, 4, 3, 2, 4, 3, 2, 4, 3, 2, 4, 3, 2, 4, 3, 2, 4, 3, 2, 4, 3, 2, 4, 3, 2, 4, 3, 2, 4, 3, 2, 4, 3, 2, 4, 3, 2, 4, 3, 2, 4, 3, 2, 4, 3, 2, 3	5 => [6] 37.5	5 => [6] 37.5

Figure 4.12: La fenêtre principale de service.



#	Date	Tag	Time	Alarm
0	2022/02/01 09:07:42.040 +01:00	1051AI5391	1	99
1	2022/02/01 09:08:02.040 +01:00	1051AI5392	2	-1
2	2022/02/01 09:46:40.090 +01:00	1051TI5302	3	-7
3	2022/02/01 10:41:31.040 +01:00	1051TI5309B	4	11
4	2022/02/01 10:41:32.040 +01:00	1051TI5309B	5	-11
5	2022/02/01 10:44:19.040 +01:00	1051TI5309B	6	11
6	2022/02/01 10:44:20.040 +01:00	1051TI5309B	7	-11
7	2022/02/01 10:44:24.040 +01:00	1051TI5309B	8	11
8	2022/02/01 10:44:27.040 +01:00	1051TI5309B	9	-11
9	2022/02/01 10:44:29.040 +01:00	1051TI5309B	10	11
10	2022/02/01 10:44:31.040 +01:00	1051TI5309B	11	-11
11	2022/02/01 10:44:59.040 +01:00	1051TI5309B	12	11
12	2022/02/01 10:45:00.040 +01:00	1051TI5309B	13	-11
13	2022/02/01 10:46:20.040 +01:00	1051TI5309B	14	11
14	2022/02/01 10:48:31.040 +01:00	1051TI5309B	15	-11
15	2022/02/01 10:50:15.390 +01:00	1051LIC5304A	16	5

Figure 4.13: La fenêtre du preprocessing.



Time interval

Input Dataset (.csv)

Choose File No file chosen

Send

Figure 4.14: La fenêtre paramètre

3. Conclusion

Dans ce chapitre nous avons présenté les outils utilisés pour l'implémentation de notre système qui est basé sur une étude de données (DATA MINING), Nous avons aussi présenté les différentes captures d'écrans montrant les différentes fenêtres de notre système. Afin de réaliser l'implémentation de notre logiciel, nous avons utilisé de différentes plateformes et les résultats de recherche ont été encourageants.

Conclusion générale

La complexité croissante des procédés industriels rend la supervision de ces derniers extrêmement difficile. Le flux important des alarmes peut perturber l'opérateur et donc influe négativement sur ses décisions. Pour résoudre ce problème, nous avons développé un service web pour détecter et analyser les corrélations des alarmes en utilisant des algorithmes de fouille de données. Les données des alarmes exprimées sous forme de fichier CSV constituent l'entrée de notre algorithme d'apprentissage.

Afin de valider notre approche, nous avons fait un stage pratique au niveau du complexe RA1K Sonatrach de SKIKDA. Les langages de programmation Python, JAVASCRIPT, ReactJS et electronJS ont été utilisé pour implémenter tous les algorithmes proposés et les résultats de validation sont très satisfaisante malgré la complexité du domaine. En perspectives, nous pensons que ce type d'approche est très récent et il est de plus en plus complexe. Plusieurs améliorations peuvent être apportées au système telles que l'enrichissement du service avec des bases de connaissances et l'incorporation de l'aspect temporel.

Bibliographie

- [1] D.Alexander et Gentil, Factory system, 1990.
- [2] R.Isermann, Control Engineering Practice, 1997.
- [3] M.Héctor AZUA, SYNTHÈSE DE LOIS DE SURVEILLANCE POUR LES PROCÉDÉS INDUSTRIELS COMPLEXES, 23 septembre 2002
- [4] N.OLIVIER-MAGET, SURVEILLANCE DES SYSTÈMES DYNAMIQUES HYBRIDES : APPLICATION AUX PROCÉDÉS, 12 décembre 2007
- [5] B.Michele, C.Marie-Odile, Surveillance et diagnostic de systemes dynamiques : approches complémentaires du traitement de signal et de l'intelligence artificielle
- [6] K. Schwab, The Fourth Industrial Revolution, 2016.
- [7] H.Hamad, M.Saad and R.Abed, "Performance Evaluation of RESTful Web Services for Mobile Devices," International Arab Journal of e-Technology, Vols. Vol. 1, no. No. 3, January 2010.
- [8] R.Gunawan and A.Rahmatulloh, "Implementasi Web Service pada Sistem Host-To-Host Pembayaran Biaya Akademik," Setrum: Sistem Kendali-Tenaga-Elektronika-Telekomunikasi-Komputer, vol. 7, no. 2, pp. 320-329, 2019.
- [9] B.BOULAHIA, « Conception et réalisation d'une application à base des services web d'une agence de voyage en ligne », mémoire de fin d'étude, Université Abderrahmane Mira de Béjaia, 2017.
- [10] A.ANISS, Composition de services web par appariement de signatures, Janvier 2008
- [11] S.Amadouni, "web services composition in uncertain environments", Doctoral dissertation, 24 septembre 2015.
- [12] A. Fron, « Architectures réparties en JAVA », Livre, 2ème édition DUNOD, Paris, 2012, ISBN 978-2-10-058217-4.

- [13] R. T. Fielding, Architectural Styles and the Design of Networkbased Software Architectures, University of California, Irvine, 2000.
- [14] H.Souraya, Thèse de Doctorat : Une approche basée agent mobile pour le mservice web sémantique, Université Mohamed Khider – BISKRA, 2014.
- [15] N.Eric ; undersetnding web service - XML, WSDL, SOAP and UDDI.independent technologies guides; 2002.
- [16] D. D. Rathod, "PERFORMANCE EVALUATION OF RESTFUL WEB SERVICES AND SOAP / WSDL WEB SERVICES," International Journal of Advanced Research in Computer Science, vol. 8, no. 7, pp. 415-420, 2017.
- [17] R. K. Sungkur and S. Daiboo, "SOREST, A Novel Framework Combining SOAP and REST for Implementing Web Services," in The Second International Conference on Data Mining, Internet Computing, and Big Data (BigData2015), 2015.
- [18] A. Memeti, F. Imeri and B. Cico, "REST vs. SOAP: Choosing the best web service while developing in-house web applications," Journal of Natural Sciences and Mathematics of UT, vol. 2, no. 3, pp. 63-68, 2017.
- [19] V. Kumari, "Web Services Protocol: SOAP vs REST," International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), vol. 4, no. 5, 2015.
- [20] B.R. Mehta, Y.J. Reddy, in Industrial Process Automation Systems, 2015
- [21] ANSI/ISA–18.2–2009 Management of Alarm Systems for the Process Industries Approved 23 June 2009
- [22] S.Basu, in Plant Hazard Analysis and Safety Instrumentation Systems, 2017
- [23] C.Jim, in Machine Learning and Data Science in the Oil and Gas Industry, 2021
- [24] A.Belakbir et N.Benkhelifa "automatisation d'une chaudière", université de Bejaia juin 2007.
- [25] D.Hoang Q.Anh , « Analyse et justification de la sécurité de systèmes robotiques en interaction physique avec l'humain », Doctoral dissertation, 2015.
- [26] M.AZZOUG, « Modélisation d'une chaudière à l'aide d'un outil graphique », Mémoire de Master, UNIVERSITE FERHAT ABBAS –SETIF, 2014

- [27] I.Boulama, « l'arrêt d'urgence de la chaudière sous une plateforme TRICONEX », Mémoire master, Université 20 AOUT 1955 SKIKDA, 2016.
- [28] Functional description control and automation (DCS)
- [29] S.KHERRAF, « COMPORTEMENT ELECTROCHIMIQUE DE L'ACIER A105 DANS DIFFERENTS MILIEUX AQUEUX », Mémoire de magister, L'université de SKIKDA, 2008
- [30] K.Radhia, K.Abdelmalek, B.Karim, B.Abderazak, Journal of Loss Prevention in the Process Industries, Volume 72, September 2021.
- [31] M.Noda, F.Higuchi, T.Takai, H.Nishitani. Event correlation analysis for alarm system rationalization. Asia-Pac J Chem Eng 2011.
- [32] H.Gao, Y.Xu , Q.Zhu, Spatial interpretive structural model identification and AHP-based multimodule fusion for alarm root-cause diagnosis in chemical processes. Ind Eng Chem Res 2016;55(12):3641–58.
- [33] Goutte and Gaussier, A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation (2005)
- [34] L.Frederic, « Microsoft Launches Visual Studio Code, A Free Cross-Platform Code Editor for OS X, Linux And Windows », TechCrunch, 29 avril 2015
- [35] <https://code.visualstudio.com/docs>
- [36] <https://nodejs.org/en/docs/>
- [37] <https://fr.reactjs.org/docs>
- [38] <https://expressjs.com/fr/>
- [39] <https://eslint.org/docs/about/>
- [40] <https://www.anaconda.com/what-is-anaconda/>
- [41] D.Christine, Anaconda for R users: SparkR and rBokeh, 1er février 2016
- [42] <https://conda.io/en/latest/>
- [43] <https://docs.python.org/>
- [44] <https://numpy.org/doc/>
- [45] <https://pandas.pydata.org/docs/>

- [46] <https://docs.python.org/3/library/json.html>
- [47] <https://matplotlib.org/>
- [48] <https://flask.palletsprojects.com/en/1.1.x/foreword/>
- [49] <https://seaborn.pydata.org/>