

Remerciements

Je tiens tout d'abord à remercier l'entreprise *Aldebaran* de m'avoir accueilli et offert l'opportunité de réaliser mon stage ingénieur au sein de l'équipe "Qualification Hardware Pepper".

Je remercie tout particulièrement M Emmanuel Nalepa de m'avoir permis de participer à ce projet captivant et formateur, m'ayant ainsi permis d'étoffer mes connaissances dans plusieurs disciplines. Sa disponibilité et sa pédagogie ont été des atouts essentiels à mon enrichissement technique.

J'adresse également mes remerciements à toutes les personnes qui m'ont proposé leur aide durant ces vingt semaines : M. Angelica Lim qui m'a fournie de précieuses informations, notamment dans le domaine du Machine Learning.

Je tiens enfin à remercier l'équipe "Qualification Hardware Pepper" pour l'aide qu'elle m'a apporté et sa très bonne humeur.

Résumé

Mon stage ingénieur, réalisé dans le cadre de ma cinquième année de formation à l'École Nationale d'Ingénieurs de Brest (ENIB), s'est déroulé au sein du département "Qualification Hardware Pepper (QWP) de l'entreprise Aldebaran, à Paris. L'entreprise s'est fait connaître dans le monde des nouvelles technologies et de la robotique humanoïde grâce au développement du robot "Nao", à destination des universités et laboratoires de recherche. La société cherche aujourd'hui à étendre son marché aux entreprises et aux particuliers avec le développement d'un tout nouveau produit : "Pepper".

Cette extension du marché s'accompagne d'une montée en puissance de la production des robots "Pepper" qui nécessite d'être accompagnée d'une nouvelle génération d'outils de production et post-production. Un des dispositifs mis en place est le "Filtering Test" : À la fin de la chaîne de production, les robots sont soumis à plusieurs tests visant à mettre à l'épreuve les différentes parties mécaniques et électroniques. Lorsqu'une erreur est détectée, les différentes données du robot sont enregistrées (e.g. température des fusibles, valeurs de l'accéléromètre, etc.). Afin de déterminer qu'elles sont les causes qui ont entraîné l'apparition de l'anomalie sur le robot, chaque donnée est étudiée minutieusement et des hypothèses sont émises. Cette tâche dite d'investigation peut s'avérer laborieuse, le souhait d'automatiser ce processus est donc important.

Le but de ma présence au sein d'Aldebaran est donc de répondre à ce besoin. En s'appuyant sur l'utilisation de méthodes d'apprentissage automatique (en anglais, on parle de Machine Learning), j'ai donc mis au point un algorithme capable de déterminer automatiquement (après une phase d'apprentissage) les causes ayant entraîné l'apparition d'anomalies sur Pepper. La mise au point de cet outil a été réalisée en trois temps :

1. auto-formation à l'apprentissage automatique et de maîtrise des outils.
2. conception et développement de l'algorithme
3. industrialisation du produit, c'est à dire en simplifier l'utilisation et le robustifier.

Le rapport de stage ci-joint présente les différentes recherches effectuées ainsi que les travaux réalisés pour répondre au mieux à la problématique initiale.

Sommaire

1	Entreprise	5
1.1	Histoire	5
1.1.1	l'Aire Nao	5
1.1.2	La famille Aldebaran	5
1.2	Les produits	5
1.2.1	Nao	6
1.2.2	Pepper	6
1.2.3	Roméo	7
1.2.4	Le système d'exploitation NAOqi	7
1.2.5	Plateforme de développement	8
2	Introduction	9
2.1	Expression du besoin	9
2.1.1	Présentation du produit	10
2.1.2	Hierarchisation des erreurs	10
2.1.3	Exemple d'analyse d'un fichier	10
2.2	Solution proposée	10
3	Le Machine Learning	12
3.1	Généralités sur le Machine Learning	12
3.1.1	Définition	12
3.1.2	Principes généraux	13
3.1.3	Les données	13
3.1.4	La décision	13
3.1.5	Le modèle	13
3.2	Les différents algorithmes	13
3.2.1	La regression logistique	13
3.2.2	Les réseaux neuronaux	13
3.2.3	SVM - Support Vector Machine-	13

3.2.4	Comparaison des algorithmes	13
4	Utilisation du Machine Learning pour l'analyse d'incidents	14
4.1	Achitecture High Level du système proposé	14
4.2	Solutions techniques testées	14
4.3	Solution technique proposée	14
4.4	Dimensionnement de la solution	14
5	Industrialisation du produit	15
5.1	Définition du terme d' "industrialisation"	15
5.2	Mise en place du process fonctionnel	15
5.3	Présentation des outils	15
5.3.1	API	15
5.4	Outils graphiques	15
6	Conclusion	16

Table des figures

Liste des tableaux

1.1	Caractéristiques technique de Nao	6
1.2	Caractéristiques technique de Pepper	7
1.3	Caractéristiques technique de Roméo	8
2.1	Déroulement d'un Filtering test	9

Chapitre 1

Entreprise

1.1 Histoire

Aldebaran (anciennement Aldebaran Robotics) est une société Française de robotique humanoïde, fondée en 2005 par Bruno Maisonnier.

1.1.1 l'Aire Nao

Constituée au départ d'une équipe de douze collaborateurs, la toute jeune entreprise se fixe comme objectif de développer des robots humanoïdes et de les commercialiser au grand public, en tant que "nouvelle espèce bienveillante à l'égard des humains" . Après trois années de recherche et développement, la société dévoile en 2008 son tout premier robot : Nao. La participation du robot humanoïde à divers évènements internationaux (e.g. RoboCup, Exposition Universelle de Shanghai en 2010, etc.) participe à sa popularisation auprès des laboratoires de recherche, des universités et des développeurs. Une seconde génération de robot Nao apparait en 2011 (dit Nao Next Gen). L'entreprise dévoile durant la même période un nouveau projet, en partenariat avec différents acteurs de la recherche, qui vise à créer un véritable robot d'assistance à la personne, Roméo.

1.1.2 La famille Aldebaran

Durant l'année 2012, Aldebaran est racheté par Softbank, société spécialisée dans le commerce électronique au Japon. Débute alors la conception d'un tout nouveau produit, le robot humanoïde Pepper. Dévoilé au grand public en 2014, il est dans un premier temps vendu au Japon auprès des entreprises. Les premiers clients à en bénéficier seront les magasins de téléphonie mobile du groupe SoftBank. Les ventes s'ouvrent dans un second temps au particuliers Japonais. La société poursuit à présent le développement de ses trois produits afin de les améliorer et de conquérir de nouveaux marchés (en Europe , en Chine et aux États-Unis).

1.2 Les produits

Aldebaran commercialise à ce jours deux produits : Nao et Pepper. Le robot Roméo est une plateforme de recherche.

1.2.1 Nao

Nao est un robot humanoïde de 58 cm de hauteur. Son public cible est principalement les laboratoires de recherche et le monde de l'éducation (allant des écoles primaires aux universités). Il est actuellement le produit le plus connu de l'entreprise auprès du grand public.

Caractéristiques techniques Caractéristiques techniques de la dernière version de Nao (V5, Evolution). 1.1

Caractéristiques générales	
Dimensions	574 x 311 x 275 mm
masse	5,4 kg
Degrés de liberté	25
Processeur	Intel Atom Z530 1.6 GHz RAM : 1GB Mémoire flash : 2GB Micro SDHC : 8 GB
Système d'exploitation	Middleware Aldebaran NAOqi basé sur un noyau Linux
Connectivité	Wi-Fi, Ethernet, USB
Batterie	Autonomie : 90 minutes en usage normal Energie : 48.6 Wh
Vision	Deux caméras frontales 2D, 1220p, 30ips
Audio	Sortie : 2 haut-parleurs stéréo 4 microphones directionnels moteur de reconnaissance vocale Nuance
Capteurs	2 capteurs infra-rouges, résistance sensible à la pression, centrale inertielle, 2 systèmes sonars, 3 surfaces tactiles

TABLE 1.1 – Caractéristiques techniques de la dernière version commerciale de Nao

1.2.2 Pepper

Dernier né d'Aldebaran, le robot Pepper est conçu pour vivre au côté des humains. Imaginé au départ pour accompagner et informer les clients dans les magasins de téléphonie du groupe japonais SoftBank, l'entreprise cherche à présent à placer son produit chez les particuliers. Le robot se base sur la structure software et hardware de Nao. Cependant, contrairement à son compagnon Nao, celui-ci se déplace non pas grâce à une paire de jambes, mais via trois roues omnidirectionnelles, facilitant son déplacement. A noter également que Pepper est équipé d'une tablette tactile sur son torse afin de faciliter les interactions Homme-Machine.

Caractéristiques techniques Caractéristiques techniques de la dernière version commerciale de Pepper (V1.6). 1.2

Caractéristiques générales	
Dimensions	1210 x 480 x 425 mm
masse	28 kg
Degrés de liberté	17
Processeur	Intel Atom E3845 1.91 GHz RAM : 4 GB Mémoire flash : 8 GB MICRO SDHC : 16Go
Système d'exploitation	Middleware Aldebaran NAOqi, basé sur un noyau Linux
Connectivité	Wi-Fi, Ethernet, USB
Batterie	Énergie : 795 Wh
Vision	2 caméras 2D 1 caméra 3D
Audio	3 microphones directionnels moteur de reconnaissance vocale Nuance
Connectivité	Wi-Fi, Ethernet
Capteurs	6 lasers, 2 capteurs infra-rouges, 1 système sonar, résistance sensible à la pression, 2 centrales inertielles, 3 surfaces tactiles

TABLE 1.2 – Caractéristiques techniques de la dernière version commerciale de Pepper

1.2.3 Roméo

Roméo est un nouveau type de robot d'accompagnement et d'assistance à la personne. Cette plateforme de recherche est soutenue par Aldebaran, ainsi que d'autres partenaires universitaires et de la recherche (e.g. INRIA, LAAS-CNRS, ISIR, ENSTA, Telecom, etc.). Aujourd'hui encore au stade de prototype, Roméo sert principalement de plateforme de tests pour les prochaines innovations majeurs d'Aldebaran (e.g. les yeux mobiles, le système vestibulaire, etc.).

Caractéristiques techniques Caractéristiques techniques de la dernière version commerciale de Roméo (V2). 1.3

1.2.4 Le système d'exploitation NAOqi

NAOqi est le système d'exploitation commun aux robots d'Aldebaran. Il se base sur la distribution de Linux Gentoo et contient plusieurs API's afin de commander et contrôler les robots.

NAOqi Core : Gestion de l'ensemble des fonctions de base des robots (e.g. mémoire, "autonomous Life", comportements du robots, etc.).

NAOqi Motion : Gestion des mouvements des robots.

NAOqi Audio : Gestion de la partie audio des robots.

NAOqi Vision : Gestion de la partie vidéo des robots

NAOqi People Perception : Ce module est utilisé pour étudier les personnes présentes autour du robot.

NAOqi Sensors : Gestion de l'ensemble des senseurs équipant les robots.

Caractéristiques générales	
Hauteur	1467 mm
masse	37 kg
Processeur	Intel ATOM Z530 1.6 GHz RAM : 1 GB Mémoire flash : 2 GB MICRO SDHC : 8 Go
Système d'exploitation	Middleware Aldebaran NAOqi, basé sur un noyau Linux
Connectivité	Wi-Fi, Ethernet
Batterie	Énergie : 795 Wh
Vision	4 caméras 2D 1 caméra 3D
Audio	3 microphones directionnels moteur de reconnaissance vocale Nuance
Connectivité	Wi-Fi, Ethernet
Capteurs	6 lasers, 2 capteurs infra-rouges, 1 système sonar, résistance sensible à la pression, 2 centrales inertielles, 3 surfaces tactiles

TABLE 1.3 – Caractéristiques techniques de la dernière version de Roméo

1.2.5 Plateforme de développement

Les robots sont fournis avec une plateforme de développement.

Choregraphe : Il s'agit d'un outil de programmation graphique basé sur une interface prenant la forme de schémas bloc. Il permet de façon simple d'interagir avec le robot et de concevoir des applications pour le robot. Il comporte également un environnement de simulation 3D permettant au développeur de tester son application sans même posséder un robot. Le logiciel permet également de disposer d'un retour sur ce que le robot perçoit (e.g. vidéo issues des caméras, données des moteurs, etc.)

Kit de développement (SDK) : Il permet de développer des applications pour les robots via plusieurs langages de programmation : C++, Python et Java.

Chapitre 2

Introduction

2.1 Expression du besoin

L'extension du marché visée par Aldebaran pour Pepper s'accompagne d'une montée en puissance de la production. Afin de la guider, des outils de vérification des produits en fin de ligne de production sont mis en place. Parmi celui-ci le "Filtering Test", ce test de vingt heures vise à stresser l'ensemble des parties mécaniques du robot afin de faire ressortir d'éventuels problèmes ou erreurs sur celui-ci. Si une anomalie survient lors du déroulement du test, différentes données relatives à l'état des systèmes mécaniques et électroniques du robot sont enregistrées dans un fichier journal (e.g. température des fusibles, valeur des accéléromètres, etc.). Afin d'identifier les causes de l'apparitions de problèmes sur Pepper, un certain nombre d'hypothèses sont émises à partir des données en sortie du système.

2 tests de 10 heures chacun, soit 20 heures de tests			
Activité	Durée (en minutes)	Temps cumulé (en minutes)	Description
Test 1 : succession de divers actions effectué 40 fois durant les 10 heures			
Intro, Danse, Outro	6 à 7	6 à 7	Réalisation des mouvements d'introduction et d'outro
Cycles de WakeUp, Rest	3 à 4	10	Passage successif en mode autonome et veille
Faux dialogues	2	12	Réalisation des mouvements effectués lors de dialogues
Rest	3	15	Réalisation des mouvements lors du passage en mode Rest
Test 12 : enchainement de 8 danses répété 5 fois durant les dix heures			

TABLE 2.1 – Déroulement d'un Filtering test

2.1.1 Présentation du produit

Afin de mieux cerner les différents problèmes techniques que peut rencontrer le robot, une description succincte de l'architecture mécanique et électronique est donnée dans cette partie.

2.1.2 Hiérarchisation des erreurs

Afin de gérer au mieux les anomalies, celles-ci sont hiérarchisées en deux catégories.

Error name : Il s'agit des erreurs visibles, c'est à dire de la conséquence liée à une anomalie hardware ou software. Cela peut correspondre par exemple à la chute du robot.

root cause : Il s'agit de l'anomalie en elle même, c'est à dire de la cause ayant entraîné l'apparition d'une "error name". Si l' "error name" est la chute d'un robot, la "root cause" peut par exemple correspondre à la détérioration d'un engrenage de la hanche.

En suivant la logique exprimée par ces définitions, une "error name" sera constituée d'une ou plusieurs "root cause".

2.1.3 Exemple d'analyse d'un fichier

Observation : — Lors du déroulement du test, le robot tombe à $t = 16972$ secondes, soit lorsque le robot réalise une séquence de mouvements particulière appelée "Heat Behavior". Les valeurs retournées par l'accéléromètre selon l'axe Z attestent de cette chute.

- On analyse alors les données liées aux systèmes mécaniques et électroniques du robot pouvant avoir une relation directe ou indirecte avec sa chute. Lorsque l'on étudie la vitesse de rotation du moteur de la hanche, on remarque qu'aux environs de $t = 16970$ secondes (c'est-à-dire 2 secondes avant la chute du robot), l'information fournie par le capteur ne suit plus la commande envoyée au moteur.
- On remarque également une augmentation anormale du courant dans le moteur de l'articulation.

Hypothèse émise : Lors de l'exécution de l'animation "Heat Behavior", le robot est amené à réaliser des mouvements amples au niveau de sa hanche entraînant un certain stress sur cette partie mécanique. Lorsque l'engrenage de la hanche arrive près de sa butée mécanique, celui-ci ne parvient pas à atteindre sa position zéro. Ce phénomène occasionne une augmentation du courant délivré dans le moteur de l'articulation ce qui entraîne le passage en mode protégé du robot, désactivant sa rigidité. Sans cette rigidité, le robot tombe ("error name"). Une étude plus poussée nous apprendra que la "root cause" du problème correspondait à une dent cassée sur l'engrenage de la hanche.

2.2 Solution proposée

De part la quantité d'informations à analyser, cette tâche peut rapidement devenir rébarbative, d'où le souhait d'automatiser ce processus d'investigation. De part la variabilité des types de données, on ne peut réduire le nombre d'information à analyser en recherchant des caractéristiques communes (e.g. moyenne, écart type, etc.). On se propose donc d'utiliser des approches algorithmiques plus poussées via des méthodes d'apprentissages automatiques (plus connues sous le terme anglais de Machine Learning). La multiplicité

des modèles composant l'apprentissage automatique nous permettra de répondre au mieux à la problématique.

Chapitre 3

Le Machine Learning

3.1 Généralités sur le Machine Learning

Le machine learning (traduire par apprentissage automatique) constitue une branche de l'intelligence artificielle. Son statut de subdivision ne renseigne cependant en rien sur l'étendu des notions contenues dans cette matière scientifique. Son champ d'étude est vaste et en perpétuelle évolution. Les solutions offertes par cette discipline permettent d'étudier toute sortes de données et d'automatiser une multitude de systèmes. L'apprentissage automatique rencontre un succès croissant qui est corrélé avec l'essor des nouvelles technologies et l'automatisation de l'analyse de volumes conséquents de données utilisateurs (Big Data). Les applications sont donc multiples, en voici quelques exemples :

- Algorithmes des moteurs de recherches (Google Deep Dream, Google TensorFlow)
- Analyse boursière
- Analyse de rapports d'erreurs
- Reconnaissance vocale, biométrie, reconnaissance d'écriture
- Robotique (vision, mouvements, prise de décision, etc.)
- Neurosciences

3.1.1 Définition

Le champ d'étude et d'application du Machine Learning étant immense, on propose de définir cette notion pour l'adapter à la résolution de notre problématique, c'est-à-dire d'automatiser l'analyse d'incidents révélés lors du filtering test. On offre ici deux définitions de l'apprentissage automatique : une première dite "High Level" qui le caractérise de manière générale , une seconde qui reflète sa dimension algorithmique.

High Level : Le Machine Learning permet à un système d'évoluer grâce à un processus d'apprentissage et ainsi de remplir des tâches qu'il est difficile, voir impossible, de remplir par d'autres moyens algorithmiques plus classiques.

Mathématique Le Machine Learning fourni les outils pour prédire une/des donnée(s) de sortie Y à partir des données d'entrée X via un processus d'apprentissage.

De nombreuses autres définitions existent pour définir ce qu'est le Machine Learning, mais elles ne correspondent pas à la dimension recherchée dans le cadre de ce projet.

3.1.2 Principes généraux

Au regard des deux définitions stipulées précédemment 3.1.1

3.1.3 Les données

3.1.4 La décision

3.1.5 Le modèle

3.2 Les différents algorithmes

3.2.1 La regression logistique

3.2.2 Les réseaux neuronaux

3.2.3 SVM - Support Vector Machine-

3.2.4 Comparaison des algorithmes

Chapitre 4

Utilisation du Machine Learning pour l'analyse d'incidents

4.1 Achitecture High Level du système proposé

4.2 Solutions techniques testées

4.3 Solution technique proposée

4.4 Dimensionnement de la solution

Chapitre 5

Industrialisation du produit

5.1 Définition du terme d' "industrialisation"

5.2 Mise en place du process fonctionnel

5.3 Présentation des outils

5.3.1 API

5.4 Outils graphiques

Chapitre 6

Conclusion