

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Université A/Mira de Béjaïa
Faculté des Sciences Exactes
Département d'Informatique

MÉMOIRE DE MASTER RECHERCHE

Filière

Informatique

Option

Réseaux et Systèmes Distribués Intelligence Artificielle

Thème

Authentification biométrique d'un chauffeur basée sur une approche comportementale

Présenté par : M^{lle} MEKAOUICHE Sirina
M^{lle} HASSANI Amal

Soutenu le 02 Octobre 2020 devant le jury composé de :

| | | | |
|--------------|--------------------|-------------------|-------------------|
| Président | Dr AMROUN Kamel | Maître de conf. A | U. A/Mira Béjaïa. |
| Examineur | Dr MOKETFI Mohand | Maître ass. A | U. A/Mira Béjaïa. |
| Promoteur | M. AISSANI Sofiane | Maître de conf. | U. A/Mira Béjaïa. |
| Co-Promoteur | M. KACIMI Farid | Maître de conf. | U. A/Mira Béjaïa. |

Année Universitaire 2019/2020

Remerciements

AMAL

Ce travail est l'aboutissement d'un dur labeur et de beaucoup de sacrifices ; mes remerciements vont d'abord à mes deux parents Nourredine et Rachida qui m'ont accompagnés durant ce long trajet et qui m'ont assisté durant ses obstacles. Ensuite, Je remercie mon encadrant Mr Sofiane AISSANI et mon Co-encadrant Mr Farid KACIMI pour avoir été compréhensibles et patients. Je remercie également tous les membres de ma famille, et plus précisément mon frère Amine et mes sœurs Lina, Milissa, Maylisse et mon petit ange Amina qui m'ont toujours soutenus et poussés à continuer mes études. Je tiens aussi à remercier mes grands-parents pour tous les encouragements et l'espoir qu'ils m'ont donné, également ma tante et mon oncle et leurs enfants Ahlem, Meriem, Walid, Idir, Hiba et la petite Asma. Je souhaite également remercier mes amis Sofiane, Sirina, Imene, Khalida, Oussama et juba. Ce présent travail a pu voir le jour grâce à leur soutien. Enfin, je remercie toutes les personnes qui ont contribué de près ou de loin à ma réussite.

SIRINA

La réalisation de ce mémoire a été possible grâce au concours de plusieurs personnes à qui je voudrais témoigner toute ma reconnaissance.

Je voudrais tout d'abord adresser toute ma gratitude à ma mère Khalida, mon père Hamza pour m'avoir encourager depuis ma naissance.

Je voudrais également être reconnaissante à mes deux tantes Razika et Hadjira sans oublier mes frères Walid et Badas et mon petit ange, la lumière de ma vie Emilie.

Je voudrais aussi exprimer ma reconnaissance envers mes amis Mahmoud, Imene, Zazak sans oublier ma binôme et amie Amel, qui m'ont apporté leur support moral.

Je remercie mes oncles Samir, Lyazid, Snoussi, Abd slam, Yasser et ma chère Grand mère Enfin, je tiens à remercier tous ceux qui m'ont aidé et participé de près ou de loin à mon parcours.

Dédicaces

Je dédie mon travail à mes
parents et ma deuxième maman, les piliers de
toute ma vie HASSANI Nourredine ,BOUFADENE
Rachida et BOUFADENE Rebiha et à tout les
malades du cancer.Ma tante était forte, souriante,
attentionnée, elle m'avait promis d'être présente à
ma soutenance alors qu'elle était mourante. C'est
évident que je lui dédie ma réussite.Ma mère
est l'ange gardien qui me protège de tout
mal, qui m'offre tout ce qu'elle a.Mon
père me pousse sans cesse ni
perte d'espoir a être la
meilleure...AMEL



Dédicaces

Je dédie mon travail
à mon exemple dans cette vie la
seule personne qui accepterait de mourir
pour m'offrir la vie, la femme la
plus forte que je connaisse ma mère
Khalida et à toutes les femmes
qui combattent chaque jour
le sexisme dans la société ...SIRINA



Table des matières

| | |
|--|-----------|
| Introduction générale | 10 |
| 1 Généralités sur la théorie des graphes | 1 |
| 1.1 Introduction | 1 |
| 1.2 Graphe orienté et graphe non orienté | 1 |
| 1.2.1 Graphe non orienté | 1 |
| 1.2.2 Graphe orienté | 1 |
| 1.3 Chemin et chaîne | 2 |
| 1.4 Circuit et cycle | 3 |
| 1.5 Graphe complet ou clique | 3 |
| 1.6 Graphe connexe | 4 |
| 1.7 Représentation d'un graphe | 4 |
| 1.7.1 Liste d'adjacence | 4 |
| 1.7.2 Matrice d'adjacence | 5 |
| 1.8 Algorithmes d'appariement dans les graphes | 5 |
| 1.8.1 Algorithmes d'appariements exacts | 5 |
| 1.8.2 Algorithmes d'appariements approximatifs | 6 |
| 1.9 Conclusion | 6 |
| 2 Généralités sur le Machine Learning | 7 |
| 2.1 Introduction | 7 |
| 2.2 Apprentissage automatique (Machine Learning) | 7 |
| 2.3 Types d'apprentissage automatique | 7 |
| 2.3.1 Apprentissage automatique supervisé | 7 |
| 2.3.2 Apprentissage automatique non supervisé | 8 |
| 2.3.3 Apprentissage automatique semi-supervisé | 8 |
| 2.3.4 Apprentissage automatique par renforcement | 8 |
| 2.4 Algorithmes d'apprentissage automatique | 8 |
| 2.4.1 La méthode k plus proches voisins | 8 |
| 2.4.2 Machines à vecteurs de support (SVM) | 8 |
| 2.4.3 Arbres de décision | 9 |
| 2.4.4 Clustering (méthode de partitionnement) | 9 |
| 2.4.5 Réseaux de neurones artificiels | 9 |
| 2.5 Conclusion | 9 |

| | | |
|----------|---|-----------|
| 3 | État de l'art | 10 |
| 3.1 | Introduction | 10 |
| 3.2 | Driving behaviour modelling system based on graph construction | 10 |
| 3.3 | An HMM-Based behavior modeling approach for continuous mobile authentication | 11 |
| 3.4 | Survey on biometry for cognitive automotive systems | 11 |
| 3.5 | Functional control of users by biometric behavior features in cloud computing . | 12 |
| 3.6 | Driver identification using ear biometrics | 12 |
| 3.7 | Continuous authentication using behavioural biometrics | 13 |
| 3.8 | A Novel GMM-Based Behavioral Modeling Approach for Smartwatch-Based Driver Authentication | 14 |
| 3.9 | A New Non-Intrusive Authentication Method based on Dynamics of Driver's Upper Body Joint Angles | 14 |
| 3.10 | Graph-based representation of behavior in detection and prediction of daily living activities | 14 |
| 3.11 | Conclusion | 17 |
| 4 | Proposition | 18 |
| 4.1 | Introduction | 18 |
| 4.2 | Solution proposée | 18 |
| 4.2.1 | Modélisation du comportement | 18 |
| 4.2.2 | Exemple illustateur | 20 |
| 4.3 | Comparaison | 23 |
| 4.3.1 | Déduction du sous graphe | 24 |
| 4.3.2 | Comparaison | 24 |
| 4.4 | Conclusion | 25 |
| 5 | Indication du codage | 26 |
| 5.1 | Introduction | 26 |
| 5.2 | Environnement de travail | 26 |
| 5.3 | Outils utilisés | 26 |
| 5.3.1 | Java | 26 |
| 5.3.2 | Eclipse | 27 |
| 5.3.3 | Java Class Library | 27 |
| 5.4 | Codage | 27 |
| 5.4.1 | Récupération des données | 27 |
| 5.4.2 | Formation des graphes séquentiels | 28 |
| 5.4.3 | Calcul de probabilités | 28 |
| 5.4.4 | Calcul de temps | 28 |
| 5.4.5 | Valeurs des nœuds | 28 |
| 5.4.6 | Calcul des poids de nœuds | 28 |
| 5.4.7 | Graphe de référence | 29 |

| | |
|----------------------------|-----------|
| <i>TABLE DES MATIÈRES</i> | 7 |
| 5.5 Conclusion | 29 |
| Conclusion générale | 31 |

Liste des tableaux

| | | |
|-----|--|----|
| 4.1 | Tableau des comportements choisis. | 19 |
| 4.2 | Tableau de probabilités | 23 |

Table des figures

| | | |
|------|--|----|
| 1.1 | Graphe non orienté | 2 |
| 1.2 | Graphe orienté | 2 |
| 1.3 | Chemin et chaîne | 3 |
| 1.4 | Circuit et cycle | 3 |
| 1.5 | Graphe complet | 4 |
| 1.6 | Liste d'adjacence | 4 |
| 1.7 | Matrice d'adjacence | 5 |
| | | |
| 4.1 | Graphe du premier essai | 21 |
| 4.2 | Graphe du deuxième essai | 21 |
| 4.3 | Graphe du troisième essai | 21 |
| 4.4 | Graphe du quatrième essai | 21 |
| 4.5 | Graphe du cinquième essai | 21 |
| 4.6 | Graphe du sixième essai | 22 |
| 4.7 | Graphe du septième essai | 22 |
| 4.8 | Graphe du huitième essai | 22 |
| 4.9 | Graphe du neuvième essai | 22 |
| 4.10 | Graphe du dixième essai | 22 |
| 4.11 | Graphe général | 23 |
| 4.12 | Schéma du processus de comparaison | 24 |

Introduction générale

De nos jours, l'automatisation des tâches prend de plus en plus d'ampleur dans notre vie quotidienne, quel que soit le domaine, les smart homes, smart cities. . . , dans le but d'améliorer le niveau de vie des utilisateurs, ce qui requière une plus grande connaissance de la vie privée de ces derniers, mais cela n'est pas sans risques. Jusqu'à présent, la sécurité des véhicules est restée classique, utilisant principalement les dispositifs de verrouillage, les alarmes ou l'empreinte digitale.

Le besoin de nouveauté dans ce domaine, notamment à cause des véhicules intelligents qui communiquent de plus en plus d'informations sensible entre eux, a poussé la communauté scientifique à réfléchir à de nouvelles méthodes pour renforcer la sécurité des systèmes de transport, ce qui a donné la naissance à un nouveau domaine de recherche dénommé « Systèmes de Transport Intelligents » abrégé STI. Les systèmes de transport intelligents (STI) sont ces nouvelles technologies appliquées aux réseaux de transport pour en améliorer la gestion et l'exploitation, aussi bien que les services aux utilisateurs, et qui permettent, entre autres, d'étudier le comportement des chauffeurs, pour mieux les protéger face aux attaques ou à eux-mêmes.

Au cours des dernières années, les chercheurs se sont de plus en plus intéressés au cas de voiture intelligente et l'authentification du conducteur est la partie la plus importante et obligatoire dans ce domaine, ce qui a conduit à un énorme développement dans ce dernier. Les travaux précédents effectués sur cette thématique sont répartis en deux classes en générales : l'authentification approche biométrique, très utilisée auparavant. Cette méthode a fait ces preuves par le passé mais comme pour toutes les nouveautés, les attaquants lui ont trouvé des failles. Par exemple pour la fameuse méthode d'utilisation des empreintes digitale ou l'attaquant imite les empreintes du propriétaire et se procure les informations dont il a besoin, ou encore la reconnaissance faciale ou l'attaquant présente une photo du propriétaire pour pouvoir voler son bien. Il y a aussi, comme c'est le cas pour notre travail l'approche comportementale ou l'authentification met l'accent sur les habitudes du chauffeur. Cette méthode rarement étudiée pourrait être la solution la plus robuste qu'on pourrait utiliser, imiter un comportement s'avère être difficile.

La science a prouvé que chaque être humain a ses propres comportements dans chaque tâche effectuée dans la vie quotidienne. En se basant essentiellement sur ses habitudes à bord d'un véhicule, nous allons représenter le comportement d'un conducteur au sein de son véhicule, plus exactement sur les diverses actions que peut effectuer ce dernier dès son entrée dans la voiture jusqu'à son démarrage. Et notre solution proposée consiste à représenter ces comportements avec des graphes ; chaque essai sera représenté avec un graphe séquentiel et le graphe général sera déduit à partir de ces derniers. En fusionnant les graphes séquentiels de tous les essais d'un conducteur avec un algorithme que nous proposons nous même pour définir les probabilités, il est possible de construire un unique graphe général pour modéliser le comportement d'un

conducteur. Différents pilotes vont avoir des différents graphes généraux. Après la phase d'apprentissage un graphe sera construit et ce dernier sera comparé au graphe général du propriétaire de la voiture pour définir le degré de dissimilarité qui sera utilisé pour décider si c'est bien le propriétaire ou pas.

Généralités sur la théorie des graphes

1.1 Introduction

Les graphes sont apparus en 1736 quand EULER a étudié la possibilité de parcourir la ville de Königsberg en passant exactement une seule fois par les ponts de la ville. De là est né tout un raisonnement, qui a plus tard aider les chercheurs à modéliser leurs problématiques sous forme de schéma ayant des points (représentant un ensemble d'entités) et des lignes qui relie certains de ces points (représentant des relations entre ces points). Ce raisonnement s'est développé en théorie et fut appelée théorie des graphes, ces points furent appelés sommets et ces lignes arrêtes [1].

1.2 Graphe orienté et graphe non orienté

1.2.1 Graphe non orienté

Un graphe non orienté G tel que $G = (S, A)$ est composé de :

- Un ensemble S d'objets appelés sommets non ordonnés. Dans la figure 1.1 l'ensemble S est équivalent à $S = \{1, 2, 3, 4, 5, 6\}$.
- Un ensemble A d'arêtes dont chacune relie un couple de sommets. Dans la figure 1.1 $A = \{(1, 2), (1, 3), (1, 6), (3, 4), (3, 5), (2, 4)\}$ [2].

L'arête qui relie deux couples de sommets quelconques (s_i, s_j) est exactement la même qui relie le couple de sommets (s_i, s_j) . Deux sommets (s_i) et (s_j) sont adjacents s'il existe une arête entre ces deux sommets (le concept d'adjacence est seulement valable pour les graphes non orientés) [2].

On appelle voisinage d'un graphe non orienté, l'ensemble des sommets adjacents du graphe. C'est à dire la liste des sommets que l'on peut directement accéder depuis le sommet courant. Ce concept s'applique uniquement pour les graphes non orientés. Le degré d'un sommet est le nombre d'arêtes dont ce sommet est une extrémité [2].

1.2.2 Graphe orienté

Un graphe orienté G tel que $G = (S, A)$ est composé de :

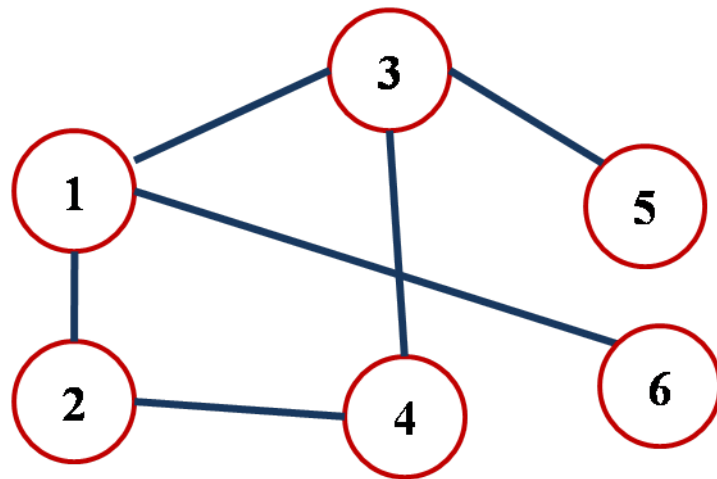


FIGURE 1.1 – Graphe non orienté

- Un ensemble S de sommets ordonnés.
- Un ensemble A d'arcs dont chacun relie un couple de sommets[2].

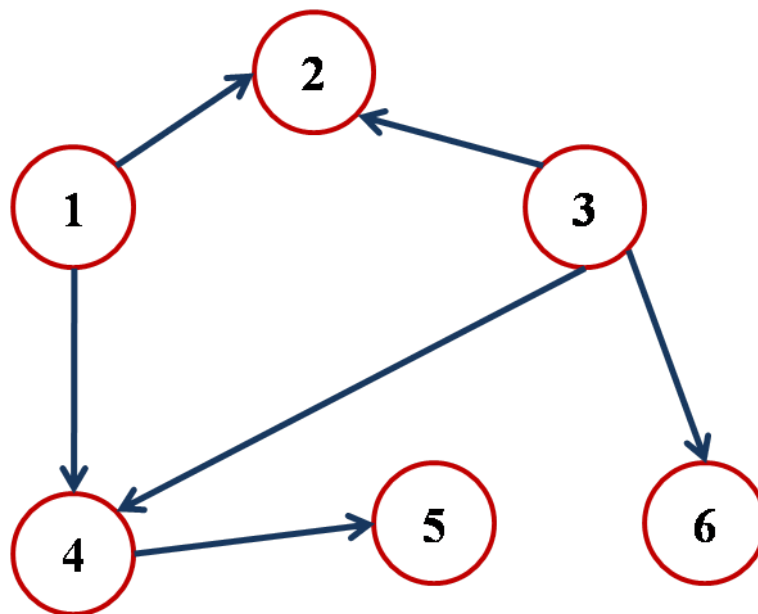


FIGURE 1.2 – Graphe orienté

1.3 Chemin et chaîne

Un chemin concerne les graphes orientés et est une suite de sommets tels que chaque sommet est relié au suivant par un arc.

La longueur du chemin est le nombre d'arcs le constituant. Par exemple le chemin représenté dans la figure 1.3 graphe de gauche : $\langle 1,4,5,2 \rangle$ a pour longueur $L=3$. Un chemin est dit élémentaire si tous les sommets qui le composent sont distincts. Par exemple dans la figure 1.3

graphe de droite le chemin $\langle 1,4,5,2 \rangle$ est élémentaire.

Une chaîne par contre concerne les graphes non orientés et est une suite de sommets tels que chaque sommet est relié au suivant par une arête[2].



FIGURE 1.3 – Chemin et chaîne

1.4 Circuit et cycle

Un circuit signifie tout simplement un chemin fermé, c'est-à-dire que le premier sommet du chemin est de même son dernier sommet. Ce terme de circuit est utilisé dans les graphes orientés. Dans le graphe représenté dans la figure 1.4 le circuit est $\langle 1,4,5,2,1 \rangle$.

Quand au cycle, c'est une chaîne fermée dont toutes les arêtes sont distinctes. Il concerne les graphes non orientés. Un graphe sans cycle est dit graphe acyclique. Dans les graphe de la figure 1.4, le cycle est $\langle 1,4,5,3,2,1 \rangle$ [2].



FIGURE 1.4 – Circuit et cycle

1.5 Graphe complet ou clique

Un graphe est dit complet si tous les sommets, pris deux à deux, sont adjacents. Dans la figure 1.5 chaque deux sommets sont adjacents[2].

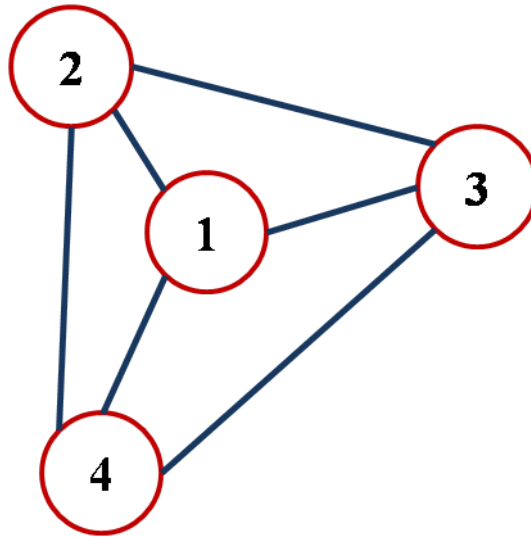


FIGURE 1.5 – Graphe complet

1.6 Graphe connexe

Un graphe est dit connexe si entre chaque deux sommets constituant le graphe, il existe une chaîne (ou chemin) les reliant. La figure 1.5 du graphe complet est un exemple de ce type de graphe.

Un graphe non connexe ne permet pas d’aller d’un sommet vers n’importe quel autre sommet distinct. Néanmoins, le graphe non connexe peut contenir une ou plusieurs composantes connexes. Une composante connexe est un sous graphe connexe[2].

1.7 Représentation d’un graphe

1.7.1 Liste d’adjacence

Une liste d’adjacence d’un graphe est une liste contenant tous les sommets du graphe et chaque case de cette liste représente le début d’une autre liste chaînée contenant les sommets qu’on peut atteindre à partir du sommet de la case départ[3].



FIGURE 1.6 – Liste d’adjacence

1.7.2 Matrice d'adjacence

La matrice d'adjacence est une matrice carrée $(M_{i,j})$ correspondante à un graphe $G=(S,A)$ avec S liste des sommets de G numérotés de 1 a n et A listes des arrêtes [4] tel que :

$$(M_{i,j}) = \begin{cases} k & \text{si il y a } k \text{ arêtes reliant } i \text{ à } j \\ 0 & \text{sinon.} \end{cases}$$

La matrice M est de taille $n*n$. Dans la figure 1.7 un exemple de graphe G (graphe précédent) et sa matrice d'adjacence :

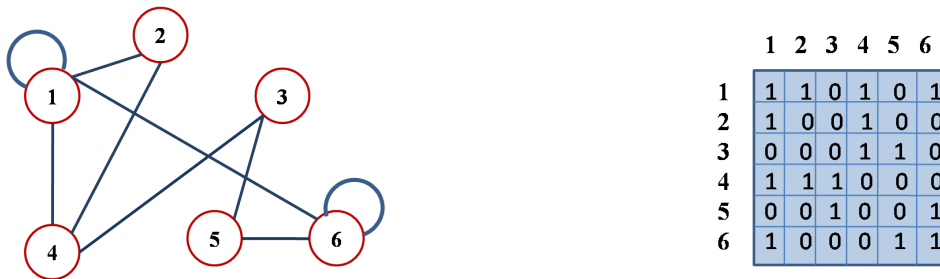


FIGURE 1.7 – Matrice d'adjacence

1.8 Algorithmes d'appariement dans les graphes

Un processus d'appariement ; aussi appelé en anglais Matching algorithm ; se fait généralement entre deux graphes $G1 = (N1, B1)$ et $G2 = (N2, B2)$ et consiste en la détermination d'une cartographie M , qui associe les nœuds du graphe $G1$ aux nœuds de $G2$ et vice versa [1].

En d'autres termes, ces algorithmes ont pour but de trouver une correspondance entre ces deux graphes afin d'en déterminer une certaine similarité[2]. Les algorithmes d'appariement se divisent en deux catégories les algorithmes exactes et les algorithmes inexacts ou approximatifs.

Cette cartographie est exprimée comme l'ensemble des paires ordonnées (n,m) avec $n \in G1$ et $m \in G2$, où chaque paire représente le mappage d'un nœud n de $G1$ avec un nœud m de $G2$. Chaque paire mi est désignée comme composante de la cartographie M [1].

1.8.1 Algorithmes d'appariements exacts

Il s'agit de trouver une fonction bijective de mise en correspondance entre les nœuds de deux graphes tout en préservant la topologie ; c'est-à-dire que chaque deux paires de nœuds adjacents du premier graphe doit correspondre a deux paires de noeuds adjacents du deuxième graphe. Il doit y avoir aussi une préservation d'étiquette, c'est-à-dire que chaque nœud du premier graphe correspond a un nœud de la même étiquette du deuxième graphe et pareil pour les arrêtes ; chaque arrête du premier graphe correspond a une arrête de la mémé étiquette du deuxième graphe[5].

1.8.2 Algorithmes d'appariements approximatifs

Les algorithmes d'appariements approximatifs ont pour première étape de générer un graphe bipartis depuis les deux graphes de bases. Puis l'enjeu est d'arriver à établir une relation entre tous les deux noeuds ; du premier graphe et du deuxième graphe. Cette relation se résume à résoudre un problème de correspondance de graphe biparti. Ce processus ressort avec un matching optimal qui sont les paires de noeuds citées auparavant. Enfin le résultat sera décidé selon le taux de correspondance entre les deux parties du matching optimal[5].

1.9 Conclusion

Dans ce chapitre nous avons expliqué quelques bases de la théorie des graphes notamment la partie qui nous intéresse le plus ; celle des algorithmes d'appariements laquelle nous allons forcément utiliser dans notre solution.

La théorie des graphes peut être utilisée dans divers domaines de la vie réelle afin de régler différents problèmes. L'enjeu est de savoir comment traduire les données liées au problème réel en un graphe simple et compréhensible. Ensuite la machine pourrait s'occuper de traiter plusieurs de ces données en un temps record et avec une solution optimale.

Généralités sur le Machine Learning

2.1 Introduction

La prise de décision automatisée a vu le jour lorsque les chercheurs ont eu comme idée d'introduire les concepts de l'intelligence artificielle à la prise de décision basique établit par les humains. Il était cependant possible qu'une entité machine puisse aider à décider en se basant sur un ensemble de données assimilés ou appris. Pour que cela deviennent concret, il fallait définir des méthodes d'aide à la décision pour les introduire à la machine. Les chercheurs ont d'abord utilisé les méthodes d'apprentissage statistiques qui nécessitaient une grande quantité de données pour aboutir à un degré d'efficacité acceptable. Ensuite il y a eu l'apparition des méthodes d'apprentissage automatique qui remédiait aux majeurs problématiques des méthodes statistiques [6].

Dans ce chapitre nous allons clarifier le concept de Machine Learning en présentant ces types et aussi en évoquant quelques méthodes d'apprentissage.

2.2 Apprentissage automatique (Machine Learning)

Le concept d'apprentissage automatique a fait sa première apparition lorsque Alan Turing a déclaré en 1947 « what we want is a machine that can learn from experience ». En 1959 le célèbre informaticien américain Arthur Samuel introduit le terme de Machine Learning en programmant un jeu de dames ou la machine s'améliorait et apprenez le jeu en jouant .

Le Machine Learning est donc une branche de l'intelligence artificielle qui consiste a ce que l'entité machine soit capable d'analyser les données observées, de mémoriser ces données puis les utiliser pour apprendre par elle-même et à de résoudre un certain problème. La machine pourra ensuite prendre une décision, c'est la solution résultante du problème [7].

2.3 Types d'apprentissage automatique

2.3.1 Apprentissage automatique supervisé

Dans le cas de l'apprentissage supervisé, le système guide l'algorithme et lui fournit des exemples complet ; entrées et résultats ; afin qu'il puisse déduire un certain raisonnement qu'il utilisera ensuite pour traiter de nouveaux cas. Chaque exemple est traité en adaptant ses paramètres de manière a ce que l'algorithme réduit au maximum la différence entre les données

résultantes obtenus et les données résultantes procurées par le système. Le taux d'erreurs diminue au fil des essais et l'algorithme pourra ensuite appliquer le raisonnement déduit sur les nouvelles entrées[8].

2.3.2 Apprentissage automatique non supervisé

L'apprentissage non supervisé se fait de manière autonome ce qui signifie que la machine sera capable de déduire les résultats par elle-même à partir des exemples qui lui ont été fournis par le concepteur. La machine traite ces données et établit des relations pour aboutir à des résultats indépendamment des résultats attendus[8].

2.3.3 Apprentissage automatique semi-supervisé

Il est utilisé dans les mêmes applications que l'apprentissage supervisé et il utilise des données étiquetées et non étiquetées pour s'entraîner. En générale Dans ce type d'apprentissage, seulement un petit nombre de données ont un label (sont étiquetées) et celles-ci sont combinées avec un grand nombre de données qui n'ont pas de label. Cela est dû au fait que les données étiquetées sont plus chères que les données sans étiquettes[8].

2.3.4 Apprentissage automatique par renforcement

L'apprentissage par renforcement est une méthode qui consiste à livrer la machine à elle-même en d'autres termes cela consiste à récompenser les décisions correctes et à pénaliser les mauvaises décisions. Cette méthode pourrait être vue comme un apprentissage non supervisé mais en guidant l'algorithme grâce aux sanctions et aux récompenses. Cet algorithme pousse l'agent intelligent à optimiser son processus de prise de décisions et à minimiser les mauvais résultats par lui-même[8].

2.4 Algorithmes d'apprentissage automatique

2.4.1 La méthode k plus proches voisins

C'est une méthode d'apprentissage supervisé dédiée à la classification son objectif est de classer les exemples non étiquetés sur la base de leur similarité avec les exemples de la base d'apprentissage. C'est l'échantillon d'apprentissage, associé à une fonction de distance et à une fonction de choix de la classe en fonction des classes des voisins les plus proches, qui constitue le modèle[7].

2.4.2 Machines à vecteurs de support (SVM)

Les machines à vecteurs de support ou Les séparateurs à vastes marges sont utilisés pour décider à quelle classe appartient un échantillon ou pour résoudre le problème de régression qui

veut dire prédire la valeur numérique d'une variable[7].

2.4.3 Arbres de décision

Méthode de classification qui représente les choix de données sous forme d'un arbre utilisée dans les domaines de l'aide à la décision de l'apprentissage automatique. Un arbre de décision est une méthode de l'apprentissage supervisé qui ne décrit pas les décisions elles-mêmes mais plutôt les données. Ce dernier est utilisé comme point de départ au processus de décision[7].

2.4.4 Clustering (méthode de partitionnement)

Technique d'exploration des données servant à résumer les informations sur les données. Cette méthode divise l'ensemble des données en classes homogènes tel que les données d'un seul cluster classe soient les plus similaires possibles[7].

2.4.5 Réseaux de neurones artificiels

Le réseau de neurones est un système qui s'inspire du schéma d'un cerveau humain. Il est divisé en plusieurs processeurs regroupés en plusieurs niveaux. Dans les réseaux de neurones, il existe une méthode d'apprentissage pour chacun des cas suivants[8] :

- Apprentissage supervisé : l'algorithme s'entraîne sur des données étiquetées jusqu'à qu'il soit en mesure de fournir les résultats souhaités.
- Apprentissage non-supervisé : le réseau de neurones analyse des données non étiquetées et s'adapte pour diminuer le taux d'erreurs.
- Apprentissage par renforcement : le réseau de neurones est récompensé pour les résultats positifs et pénalisé pour les résultats négatifs.

2.5 Conclusion

Ce chapitre nous a permis de connaître un peu plus sur le domaine d'intelligence artificielle et plus précisément sur le machine learning qui est un élément très important dans notre solution.

Le machine learning est un domaine d'actualité. Il permet de nos jours de mettre en place des systèmes qui apprennent de leurs expériences tout comme la race humaine. Cet un domaine vaste qui inclut beaucoup d'algorithmes et de sous branches. Petit à petit le monde parlera machine learning et utilisera machine learning, l'humain ne pourra plus s'en passer.

État de l'art

3.1 Introduction

Les systèmes de sécurité utilisant l'authentification biométrique ont réalisé un exploit mémorable dans l'histoire de l'authentification. Mais comme toute nouveauté qui alimente le domaine de la sécurité, il existe des failles associées à cette méthode que les pirates informatiques utilisent avec des intentions malveillantes. Au jour d'aujourd'hui, l'authentification comportementale reste un nouveau concept pour les chercheurs et pirates, c'est pour cela que ce concept est une thématique d'actualité qui vaut la peine d'être étudiée.

Dans ce chapitre, nous allons présenter différents travaux existants dans la littérature qui tourne autour de l'authentification comportementale suivis d'une comparaison entre ces articles.

3.2 Driving behaviour modelling system based on graph construction

Sei-Wang et al. ont défini une méthode de représentation du comportement d'un chauffeur basée sur les graphes. Cette représentation sera utilisée par la suite dans la phase prise de décision du système de sécurité. La méthode définit quatre étapes : l'entrée des données brutes, l'extraction des nœuds, construction des DRM (Driving Relation Map) et la construction du graphe final DHG (Driving Habit Graph). Les données brutes sont recueillies par des capteurs placés sur la voiture selon des facteurs bien définis. Ces données, qui sont sous forme d'organigramme, sont utilisées pour la deuxième étape qui est l'extraction des nœuds. La méthode veut que les facteurs ayant traversés des changements considérables pendant la période de test, soient ajoutés à la liste des nœuds du DRM. L'étape de construction des DRM repose sur l'enchaînement des événements qu'on pourrait qualifier ici de facteurs et sur le temps écoulé entre le début d'un événement et un autre. Si ce temps est considérable, l'algorithme construit un seul nœud représentant l'événement dans un seul niveau de DRM, si non il associe entre les deux événements en créant deux nœuds dans un seul niveau. Après la construction du DRM vient l'étape de la construction du graphe, les nœuds du graphe sont les DRM construits dans la troisième étape et les arcs ont un poids calculé avec la formule de probabilité qui apparaît dans l'article. Cette valeur représente la probabilité de passer d'un DRM à un autre DRM.

3.3 An HMM-Based behavior modeling approach for continuous mobile authentication

Avec la grande popularité que connaissent les mobiles tactiles de nos jours, l'authentification par mot de passe est devenue moins adaptés et plus vulnérable aux attaques. Cependant il a fallu trouver un mécanisme d'authentification qui adopte le concept du tactile et qui soit plus robuste aux attaques.

Après les méthodes d'authentification proposées auparavant, utilisant la voix de l'utilisateur ou sa méthode de frappe sur le clavier, cet article propose une méthode d'authentification continue ou active qui a pour but de vérifier l'identité de l'utilisateur non pas au moment où il veut accéder mais aussi tout au long de son utilisation du mobile tactile. Cette méthode basée sur le HMM (Hidden Markov Model), se fait en deux étapes. La première étape consiste à enregistrer le comportement de l'utilisateur concernant sa façon d'utiliser le mobile car selon les études faites à cet égard, chaque utilisateur utilise son téléphone d'une manière propre à lui. Ce enregistrement de données se fait en utilisant le modèle HMM. Ensuite dans la deuxième étape, l'algorithme reprend les données enregistrées et à chaque utilisation du mobile, il les compare avec les nouvelles données recueillies de manière continue parallèlement avec l'utilisation du téléphone.

3.4 Survey on biometry for cognitive automotive systems

Dans cet article, les auteurs proposent un modèle complet basé sur la biométrie pour la sécurité du conducteur, de son véhicule et de son entourage. Pour cela ils ont définies quatre étapes ou niveaux de sécurité.

La première étape consiste à sécuriser le véhicule intelligent pour qu'il puisse fournir au conducteur des informations concernant ce qu'il l'entoure et ainsi éviter des dommages graves. Pour cela, un simple rétroviseur ne suffit pas (il y a des angles morts), il faut donc mettre en place des capteurs de mouvement, quatre caméras de résolution, un scanner à laser et un système de localisation. Le véhicule pourra établir un modèle en 3D du véhicule et des objets qui l'entourent (autres véhicules, arbres, piétons... etc.), en déduire leurs vitesses et leurs positions et ainsi filtrer les objets en mouvements des autres objets.

La deuxième étape étant la surveillance du conducteur, comprend une section d'analyse de son comportement, pour ainsi en déduire son état mental et physique. Cette analyse a pour but de vérifier l'identité du conducteur mais aussi d'identifier si oui ou non il est en mesure de conduire sainement. À cet égard, des capteurs qui sont placés à l'intérieur et à l'extérieur du véhicule se chargent de capturer ce type d'informations, des rétroviseurs de reconnaissance se chargent d'identifier le conducteur et les caméras connectées enregistrent les vitesses d'approches des autres véhicules.

3.5 Functional control of users by biometric behavior features in cloud computing

De nos jours, le domaine du piratage et le vol de données s'est propagé partout dans le monde et a cependant évolué, il a fallu parallèlement revoir les anciennes méthodes de sécurité et d'identification pour les fortifier contre les nouvelles attaques. L'authentification biométrique est l'une des nouvelles méthodes résultante, il s'agit d'identifier l'utilisateur par ce qu'il est ou ce qu'il a. Cette est une méthode très efficace car chacun a des caractéristiques ou des comportements propre à lui.

Cet article comporte une classification complète des méthodes d'authentification biométriques mais aussi une méthode proposée après études des différentes classes. Selon l'article, on peut organiser les méthodes d'authentifications biométriques en trois classes : physiologiques, comportementaux et chimiques. La méthode physiologique est la plus vieille méthode implémentée comme le scan de l'empreinte du doigt, scan de la main, scan de la rétine de l'œil mais aussi la reconnaissance par l'iris de l'utilisateur. La méthode chimique comporte l'identification par ADN ou par sang. Quant à la méthode comportementale, elle comporte le scan de signature, la frappe du clavier (vitesse, temps d'appuie... etc.) Ou encore l'enregistrement de la voix.

La méthode proposée utilise le mot de passe et l'identifiant comme premier niveau de sécurité, puis comme deuxième niveau le comportement de l'utilisateur. Dès la première requête de connexion d'un utilisateur, le serveur enregistre ses comportements, ses goûts et ses habitudes avec une sorte de cookie et les stocke dans sa banque d'informations. Lors de la deuxième requête de connexion, l'utilisateur saisit son mot de passe et son identifiant, le serveur vérifie son identité et charge quelques questions qu'il présente à l'utilisateur. L'utilisateur va répondre aux questions et le serveur va vérifier les réponses en se basant sur les informations stockées dans la banque d'informations du serveur. Cette méthode d'authentification propose trois niveaux de sécurité avec faible coût et sans avoir à ajouter d'autres équipements pour la sécurité (client serveur). Néanmoins, la méthode est fonctionnelle qu'après la deuxième connexion pour absence des connaissances concernant le comportement des clients à la première connexion.

3.6 Driver identification using ear biometrics

Dans cet article, la méthode proposée utilise la reconnaissance par oreille du conducteur. Pour cela des caméras utilisant la thermographie sont placées à une distance de 0.5m à 1m du conducteur, celles-ci seront réglées à des paramètres précis. Le rôle de ces caméras est de mesurer la chaleur dans le champ de capture et de capturer des images qui seront transformées en thermogramme.

Le thermogramme capturé fournit des informations concernant la répartition de la température à la surface de l'oreille du conducteur. Un algorithme pour la détection de la forme de l'oreille est appliqué et va tout d'abord normaliser le thermogramme en une taille adéquate. Le détecteur utilisé est le détecteur CANNY qui détecte les bords de l'oreille. Il est utilisé pour sa

fiabilité et car il utilise la convolution gaussienne avec les images et les dérivations ultérieures.

Pour la reconnaissance du conducteur, un réseau de neurones artificiel similaire à celui du cerveau humain a été utilisé. Le réseau de neurones est un réseau multi couches et doit être accompagné d'un algorithme de rétro propagation adaptatif. L'algorithme fonctionne avec au moins une seule couche cachée et utilise la fonction d'activation sigmoïde. Dans l'article, le réseau résultant se compose de trois couches de neurones : couche d'entrée, couche de sortie et une couche cachée. Le nombre de couches d'entrées correspond au nombre de pixels des images normalisées .les pixels de l'image sont regroupés en unités et

on attribue chaque unité a une couche d'entrée. Le nombre de neurones dans la couche cachée a fait l'objet de plusieurs études, dans l'article le nombre est choisi heuristiquement et correspond a un nombre $X+1$ neurones, le 1 représente le neurone fictif. Quant a la couche de sortie, elle contient un nombre de neurones égal au nombre de personnes en identification. La valeur de sortie est ensuite normalisée en une probabilité selon cette équation :

- *Probabilité d'appariement* = $OUT_i / \sum OUT_i$ pour i allant de 1 jusqu'à k
- OUT_i : la valeur de sortie du neurone i
- K : le nombre de neurones activés

L'oreille du conducteur est capturée sous 5 angles. Sous un angle de 90° , la valeur seuil a été définie à 0,9. Pour les images capturées à des angles inférieurs à 80° et 100° , la valeur seuil à 0,7 et pour les angles 70° et 110° à 0,5. L'authentification réussit quand la probabilité d'appariement est supérieur au seuil.

3.7 Continuous authentication using behavioural biometrics

Pour surmonter les inconvénients des authentifications comportementales existantes, cet article propose une authentification continue basée sur trois modalités biométriques : la manière de frapper, souris dynamique et l'interaction logiciel. Cette authentification est une authentification multimodale. Chaque modalité capture les données de l'utilisateur séparément et les transmet au comparateur adapté afin de comparer les anciennes données et les nouvelles données. Le résultat de chaque comparateur est transmis à une unité de fusion .Le score final sera transmis au module supérieur et lui va décider si l'utilisateur va continuer son activité ou s'il sera exclu du système.

Ces trois modalités ne nécessitent aucun matériel supplémentaire pour la capture des données de l'utilisateur et la collecte de données se fait sans interruption du travail de l'utilisateur.

3.8 A Novel GMM-Based Behavioral Modeling Approach for Smartwatch-Based Driver Authentication

Dans cet article, les auteurs ont créé un système expert qui analyse les comportements d'un chauffeur de voiture en utilisant des capteurs intégrées dans une montre intelligente (smartwatch) portée sur la main gauche du conducteur cette montre avec ces capteurs capte les mouvements de la main du chauffeur et sa façon de tenir le volant qui diffèrent selon les scénarios de conduite. Dans cet article ils se sont basés juste sur Le comportement au volant du chauffeur qui a été modélisé par une approche de modélisation basée sur le mélange modèle gaussien GMM (Gaussian mixture model).

3.9 A New Non-Intrusive Authentication Method based on Dynamics of Driver's Upper Body Joint Angles

Dans cet article, les auteurs proposent une nouvelle approche pour l'authentification du chauffeur basé sur les comportements du haut du corps de ce dernier plus exactement sur la dynamique de ces angles articulaires du haut du corps. Ces comportements sont collectés en utilisant une caméra d'un capteur de profondeur.

La méthode d'apprentissage automatique k-NN le K plus proche voisins est utilisée comme algorithme de classification pour l'authentification du vrai propriétaire du véhicule.L'approche peut authentifier ou non le conducteur en faisant la correspondance entre les comportements de conduite du nouveau conducteur et les comportements de conduite du conducteur propriétaire du véhicule.

3.10 Graph-based representation of behavior in detection and prediction of daily living activities

Dans cet article, P. Augustyniak, G. Slusarczyk présentent une autre façon de représenter les comportements humains dans un système de surveillance dédié au suivi des activités de la vie quotidienne d'un individu, ceci pour but de détecter les événements potentiellement dangereux afin d'éviter des incidents .Les auteurs ont utilisé les graphes pour modéliser les comportements de l'individu ou les étiquettes des nœuds représentent les actions alimentaires tandis que les arcs expriment le passage dans une pose donnée de l'état de sujet.

| Référence | Méthode | Paramètres d'authentification | Matériel | Avantages | Type d'authentification | Authentification d'un chauffeur |
|-----------|--|--|------------------------------------|---|--------------------------------|-----------------------------------|
| [9] | DHG | Angle du volant, Degré d'actionnement du frein, Accélération latérale du véhicule... | Capteurs | DHG permet un système d'assistance à la conduite mieux adapté aux conducteurs | comportementale | oui |
| [10] | chaînes de markov | les motifs de défilement horizontales et verticales de l'utilisateur | téléphone | une méthode qui offre la possibilité d'être formé avec uniquement les données des utilisateurs ,peut être mise à jour avec de nouvelles données sur une période de temps. | Comportementale | non(utilisateur d'un smart phone) |
| [11] | Processus en 4 étapes | paramètres biométriques et comportementales | capteurs,caméras et GPS | non précis | Comportementale et biométrique | oui |
| [12] | Découverte de comportements et clés des caractéristiques | non précis | Ordinateur | Pas besoin de matériel supplémentaire , non coûteuse et facile à utiliser | Comportementale | non(cloud Computing) |
| [13] | Traitement d'image avec réseaux de neurones | Caméra IR | Changement de chaleur de l'oreille | non précis | Biométrique | oui |

| | | | | | | |
|------|--|---|---|---|--------------------------------|--------------------------------|
| [14] | Authentification Modale et continue | Frappe sur clavier, clics sur souris et interaction logiciel | Ordinateur | Plus robuste aux attaques due à l'authentification continue. | Comportementale et biométrique | Non (Utilisateur d'ordinateur) |
| [15] | GMM | Orientation de la montre, vitesse d'accélération de la montre | Montre intelligente | Assure également la sécurité du conducteur | Comportementale | oui |
| [16] | Modélisation des mouvements du haut du corps | Mouvement des coudes et du cou du chauffeur | Capteurs placés dans la voiture | Non précisé | Comportementale | oui |
| [17] | Modélisation à l'aide des graphes | Mouvement du corps humain | Caméra, microphone, accéléromètre et cardiofréquencemètre | Système pouvant être utilisé pour l'authentification et pas seulement pour le suivi | Comportementale | non (Individu quelconque) |

3.11 Conclusion

Dans ce chapitre, nous avons fait un tour d'horizon de ce qui a été proposé dans la littérature. Plusieurs travaux de recherche ont traité cette thématique. À partir des données collectées, nous avons pu comprendre le sujet et tracer un chemin qui nous a conduit à notre problème : l'authentification d'un chauffeur d'une voiture. Le problème ainsi que la solution que nous proposons seront expliqués en détails dans le chapitre suivant.

Proposition

4.1 Introduction

Dans ce chapitre nous allons proposer une solution à notre problématique qui est l'authentification du chauffeur d'une voiture à partir de ses comportements. Cette solution se basera sur quelques comportements sélectionnés qui représenteront les habitudes du chauffeur. Ce processus débutera dès lors que le conducteur entre dans sa voiture jusqu'à ce qu'il démarre.

4.2 Solution proposée

4.2.1 Modélisation du comportement

Pour pouvoir manipuler les comportements du chauffeur et faciliter leurs compréhension par le système, nous devons tout d'abord penser à les traduire en un langage ou une schématisation simple et comprise par la machine. De ce fait, nous avons choisi d'utiliser la théorie des graphes pour la modélisation de ces comportements sélectionnés.

L'idée consiste à créer pour chaque conducteur un graphe général (graphe de référence) qui représente ses habitudes dès son entrée dans la voiture jusqu'à ce qu'il démarre et cela se fait en plusieurs étapes :

Les comportements choisis

Comme mentionné auparavant, nous nous sommes basés sur quelques comportements seulement pour pouvoir effectuer cette étude, pour cela nous avons choisi six actions afin de définir les habitudes du conducteur :

- Réglage des rétroviseurs dans la voiture.
- Mettre la ceinture de sécurité.
- Positionnement du siège.
- Allumage de musique.
- Démarrer.
- Entrer dans la voiture.

Valeurs des données

Les entrées du le système proposé sont des données brutes obtenues à partir de plusieurs capteurs placés dans le véhicule, chaque comportement a une différente manière de capter les

valeurs lui correspondant :

- Réglage des rétroviseurs : seul le rétroviseur central de la voiture est pris en considération, le capteur calcule les degrés d'inclinaison de ce dernier en radian.
- Mettre la ceinture de sécurité : pour ce comportement la valeur est booléenne si oui ou non le conducteur met sa ceinture.
- Position du siège : trois réglages sont pris en considération pour positionner le siège du conducteur, le premier est l'inclinaison du coussin du dos qui est calculé en radian, le deuxième est le réglage de la hauteur du siège elle est calculé en centimètres et troisièmement, régler l'avancée ou le recul du siège et cela est calculé en centimètres.
- Allumage de musique : comme le deuxième comportement la valeur de ce comportement est booléenne.
- Démarrer : c'est un comportement sans valeur.
- Rentrer dans la voiture : c'est un comportement sans valeur.

Le tableau 4.1 récapitule les comportements cités en haut et les intervalles de variations de leurs valeurs :

| Num | Symboles | Comportements | Valeurs |
|-----|----------|---|--------------------------|
| 01 | Rr | Réglage des rétroviseurs | [1718.87rad, 8594.37rad] |
| 02 | Mc | Mettre la ceinture | Oui/non |
| 03 | Ps1 | Inclinaison du coussin du dos | [0rad, 10313.2rad] |
| 04 | Ps2 | Réglage de la hauteur | [0cm, 30 cm] |
| 05 | Ps3 | Réglage de l'avancée ou le recul du siège | [0cm, 45cm] |
| 06 | Am | Allumage musique | Oui/non |
| 07 | Dm | Démarrer | |
| 08 | Rv | Rentrer dans la voiture | |

TABLE 4.1 – Tableau des comportements choisis.

Création des nœuds

La construction du graphe commence par former les états, dans notre graphe les états représentent les actions du conducteur énoncés auparavant avec comme poids de nœud les valeurs capturées.

Formation des arcs

Après obtention des états, on passe aux transitions d'un état à un autre. Cette étape demande le recours aux notions de probabilités. Chaque passage respecte une certaine probabilité qu'on définira comme suit :

- Pendant la phase d'apprentissage un nombre d'essais est fixé pour définir combien de fois on pourra enregistrer le comportement du chauffeur lors de sa conduite.
- A chaque essaie un graphe séquentiel sera construit ou un état aura un seul choix de transition vers un autre état. Comme étant un enchaînement d'actions.

- Après obtention de n graphes on passe à la construction du graphe général en passant évidemment par le calcul de probabilité qui est une étape obligatoire pour définir les liaisons entre les nœuds.

Calcul des probabilités

Pour le calcul des probabilités, le système effectue plusieurs divisions selon le pseudo algorithme suivant :

Pour chaque couple de nœuds (A, B) tel que A va vers B :

Algorithm 1 Algorithme de calcul de probabilités

On parcourt tous les graphes séquentiels à partir du premier essai (premier graphe) jusqu'au dernier essai (dernier graphe) :

Soit j , s , N_{bn} et n des entiers

Soit P un réel

for $j=1$ to n **do**

if A apparaît dans le graphe séquentiel **then**

$N_{bn} \leftarrow N_{bn} + 1$

end if

if A va vers B **then**

$s \leftarrow s + 1$

end if

end for

$P = s \div N_{bn}$

Poids d'arc

L'ensemble des actions s'effectue dans un laps de temps qui diffère d'une personne à une autre. Le temps de passage d'une action à une autre diffère aussi d'un conducteur à un autre. C'est pour cette raison qu'on a choisi d'exprimer le temps sur notre graphe de modélisation. Pendant la phase d'apprentissage le temps sera calculé avec des capteurs et sera pris en considération comme étant poids d'arcs sur les graphes séquentiels et lors de la construction du graphe général on calcule le poids d'arc de cette façon :

- Soient $\{t_1, t_2, t_3 \dots t_n\}$ les différentes durées prises par le chauffeur pour passer d'une action A à une action B dès le premier essai jusqu'au nième essai.
- T le temps considéré comme poids de l'arc $T = t_1 + t_2 + t_3 \dots t_n$

4.2.2 Exemple illustrateur

Voici un exemple de construction d'un graphe général des habitudes d'un chauffeur pendant une phase d'apprentissage de 10 essais :

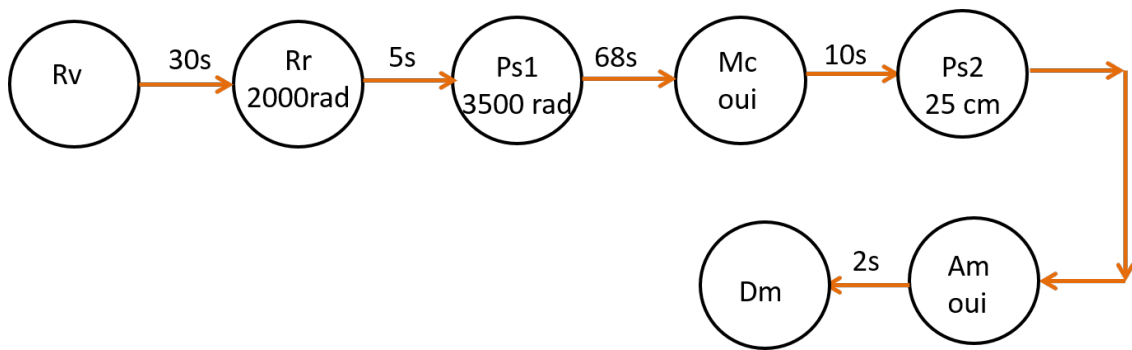


FIGURE 4.1 – Graphe du premier essai

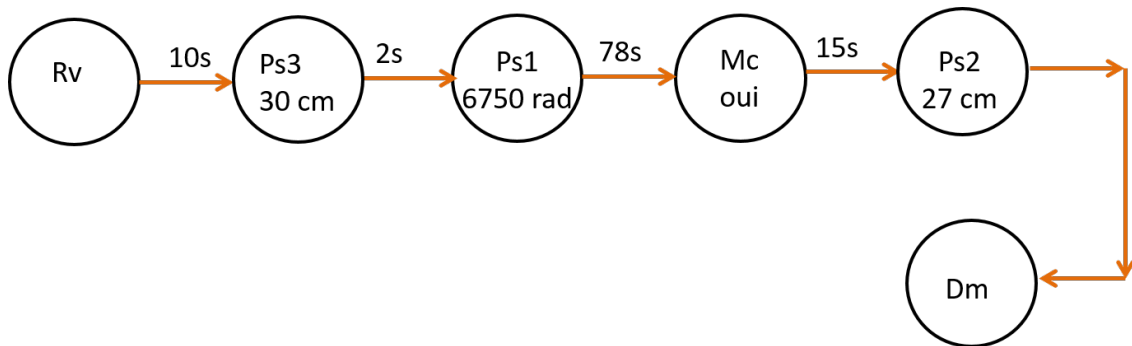


FIGURE 4.2 – Graphe du deuxième essai

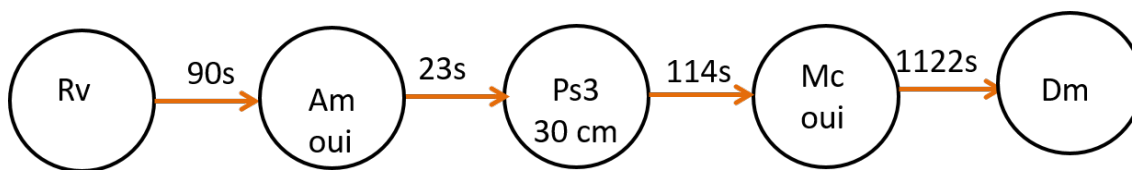


FIGURE 4.3 – Graphe du troisième essai

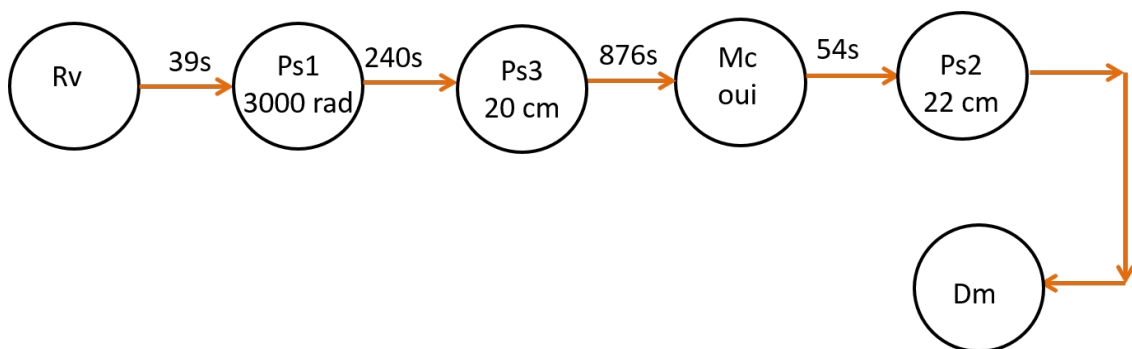


FIGURE 4.4 – Graphe du quatrième essai

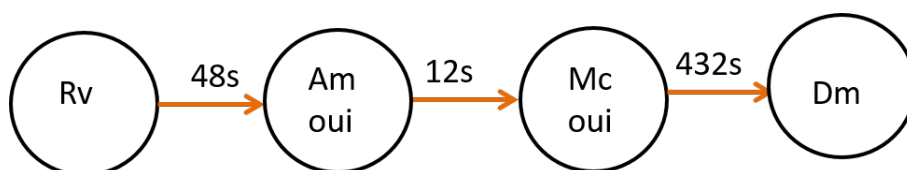


FIGURE 4.5 – Graphe du cinquième essai

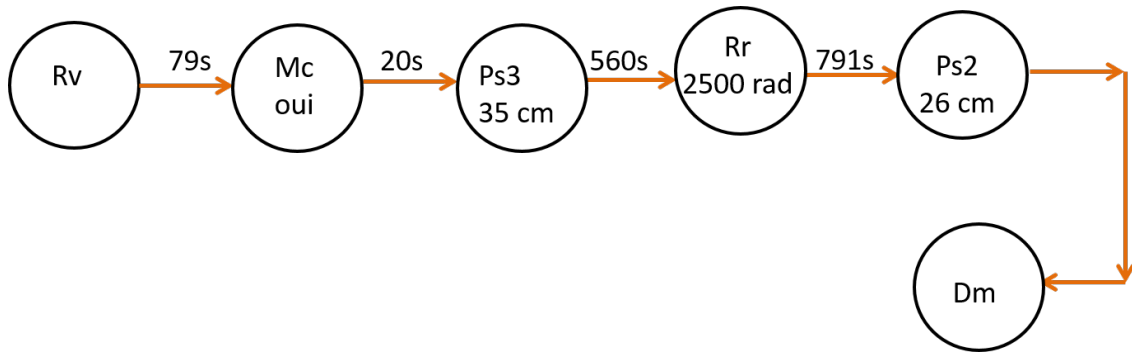


FIGURE 4.6 – Graphe du sixième essai

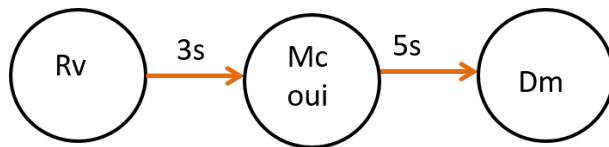


FIGURE 4.7 – Graphe du septième essai



FIGURE 4.8 – Graphe du huitième essai

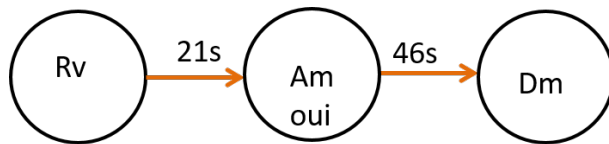


FIGURE 4.9 – Graphe du neuvième essai

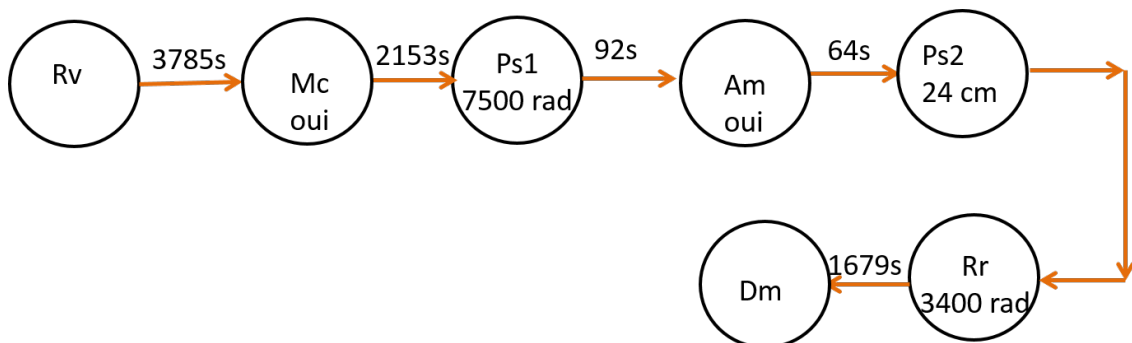


FIGURE 4.10 – Graphe du dixième essai

| - | Rv | Ps1 | Ps2 | Ps3 | Rr | Mc | Am | Dm |
|-----|----|------|------|------|------|------|------|------|
| Rv | 0 | 0.1 | 0 | 0.1 | 0.1 | 0.3 | 0.4 | 0 |
| Ps1 | 0 | 0 | 0 | 0.25 | 0 | 0.5 | 0.25 | 0 |
| Ps2 | 0 | 0 | 0 | 0 | 0.2 | 0 | 0.2 | 0.6 |
| Ps3 | 0 | 0.25 | 0 | 0 | 0.25 | 0.5 | 0 | 0 |
| Rr | 0 | 0.25 | 0.25 | 0 | 0 | 0.25 | 0 | 0.25 |
| Mc | 0 | 0.11 | 0.33 | 0.11 | 0 | 0 | 0 | 0.45 |
| Am | 0 | 0 | 0.17 | 0.17 | 0.17 | 0.17 | 0 | 0.32 |
| Dm | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

TABLE 4.2 – Tableau de probabilités

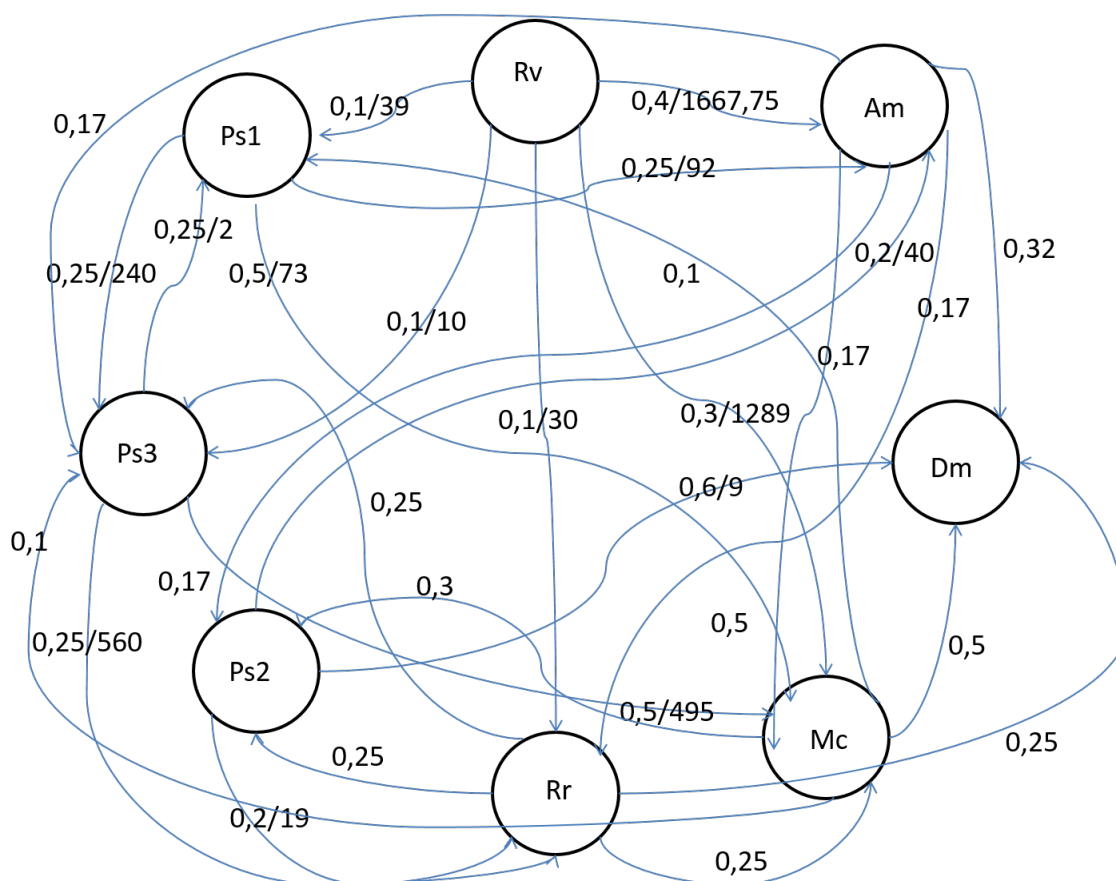


FIGURE 4.11 – Graphe général

4.3 Comparaison

Dans notre proposition, l'algorithme d'appariement entre le graphe de référence ; construit après de nombreux essais de la phase d'apprentissage ; et le premier graphe séquentiel construit pour le premier essai du conducteur après la phase d'apprentissage se décompose en deux étapes fondamentales. Le processus est expliqué à travers la figure 4.12 :

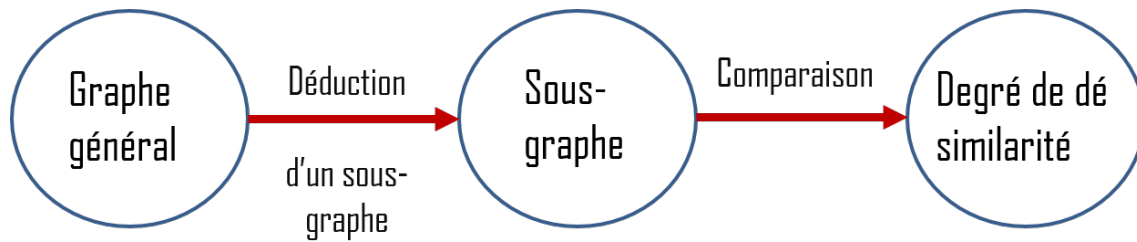


FIGURE 4.12 – Schéma du processus de comparaison

4.3.1 Déduction du sous graphe

Dans notre cas, le graphe général contient tous les nœuds et plusieurs transitions entre ces nœuds. La comparaison entre un tel graphe et un graphe séquentiel généré après la phase d'apprentissage s'avère quasiment impossible. Pour remédier à ce problème majeur, nous avons choisis d'utiliser le concept de sous graphe. L'enjeu est de déduire un graphe à partir du graphe général, contenant que les nœuds qui apparaissent dans le graphe séquentiel (après la phase d'apprentissage), ceci en gardant les mêmes poids de nœuds que dans le graphe de référence. Ensuite on suit le même raisonnement pour les arcs en gardant les valeurs des probabilités calculées dans la phase d'apprentissage pour le graphe déduit. Cela nous donnera deux graphes comparables et pourra réduire le taux d'erreur de l'algorithme.

4.3.2 Comparaison

L'étape de comparaison se décompose à son tour en deux sous comparaisons ; la comparaison des poids de nœuds et la comparaison des poids d'arcs ; chacune est à son tour expliquée dans la partie suivante :

Comparaison de poids de nœuds

Pour cette étape nous allons soustraire chaque deux poids d'un même nœud contenus dans les deux graphes à comparé. Ces soustractions nous donnerons des valeurs qu'on va ensuite diviser puis additionner pour donner une moyenne. Le résultat représente le taux de dé similarité par rapport aux nœuds.

Comparaison de poids d'arcs

De même que pour le calcul précédent, cette étape consiste à calculer les soustractions entre chaque deux valeurs du même nœud des deux graphes. Ensuite ces soustractions passeront par une étape de division et une autre étape d'addition. Ce calcul nous donne le taux de dé similarité par rapport au poids d'arc.

Une fois la partie comparaison terminée le résultat sera deux valeurs de dé similarités. Ces valeurs jugeront si ou non les deux graphes sont similaires et si oui ou non le chauffeur est le même.

4.4 Conclusion

Dans ce chapitre nous avons expliqué la solution qu'on propose pour cette problématique en plusieurs étapes. L'étape une étant de définir un format logique et numérisable des nombreux essais qu'effectue le chauffeur durant la phase d'apprentissage. Ensuite, il fallait établir un lien entre ces essais et former un graphe de référence propre au chauffeur. La dernière étape est de pouvoir déduire si oui ou non c'est le même chauffeur qui conduit la voiture après la phase d'apprentissage.

Dans le chapitre suivant, nous allons numériser la solution proposée en essayant de simuler un exemple du déroulement de ce système. Ceci pour bien expliquer ces différentes étapes et pour concrétiser notre idée.

Indication du codage

5.1 Introduction

Après avoir défini notre problématique et déterminer théoriquement sa solution et son architecture, nous nous sommes immergés dans sa mise en œuvre. Comme vu précédemment, nous nous sommes concentrés sur quelques actions confinées entre l’instant d’arrivée du chauffeur jusqu’au démarrage de la voiture. C’est l’intervalle d’actions que nous considérons le plus important pour authentifier un chauffeur. Dans ce chapitre, nous présentons l’environnement et les conditions de notre codage.

5.2 Environnement de travail

Afin de mener à bien ce projet, nous avons utilisé un ensemble de matériels dont : les principales caractéristiques sont les suivantes :

- Processeur : Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz 2.40GHz.
- RAM : 4.00 GO.
- Système d’exploitation : Windows 10 64 bits.

5.3 Outils utilisés

5.3.1 Java

Java est une technologie qui définit le langage de programmation orienté objets et qui définit aussi une plateforme informatique. C’est une technologie très utilisée dans le développement d’application. Créé par James Gosling et Patrick Naughton employés à l’entreprise Sun Microsystems en 1995, et rachetée par la société Oracle en 2009. Dans ce qui suit quelques caractéristiques de java [18] :

- Les logiciels écrits dans ce langage sont compilés vers une représentation binaire qui s’exécute dans une machine virtuelle java.
- Reprend en grande partie la syntaxe du langage C++.
- Java a donné naissance à un système d’exploitation (JavaOs), à des environnements de développement (eclipse/JDK), des machines virtuelles (MSVM,JRE) applicatives multiplate-forme (JVM), une déclinaison pour les périphériques mobiles/embarqués (J2ME), une bibliothèque de conception d’interface graphique (AWT/Swing), des ap-

plications lourdes (Jude, Oracle SQL Worksheet, etc.), des technologies web (servlets, applets) et une déclinaison pour l'entreprise (J2EE).

- Permet aux développeurs d'écrire des logiciels sur une plate-forme et de les exécuter sur pratiquement toutes les autres et aussi de combiner des applications ou des services basés sur le langage Java pour créer des applications ou des services très personnalisés.

5.3.2 Eclipse

Eclipse est un environnement de développement intégré (IDE) pour le langage java et d'autres langages. Écrit principalement en java et lancé par IBM. Sa plateforme est composée de :Platform Runtime démarrant la plateforme et gérant les plug-ins, SWT la bibliothèque graphique de base de l'EDI, JFace une bibliothèque graphique de plus haut niveau basée sur SWT et Eclipse Workbench dernière couche graphique permettant de manipuler des composants tels que des vues, des éditeurs, des perspectives...et d'autres. Voici quelques avantages d'Eclipse [19] :

- Documentation : de nombreux tutoriels et documentation du développeur existent sur Eclipse.
- Large communauté : la communauté de l'IDE Eclipse est très active et contribue à son évaluation en suivant les évolutions.
- Open source : il est gratuit.
- Large notoriété d'Eclipse IDE : il est approuvé par de nombreux développeurs dans le monde.

5.3.3 Java Class Library

- (JCL) est une bibliothèque standard du langage Java, qui contient un ensemble de bibliothèques logicielles chargées dynamiquement.
- JCL fournit aux programmeurs à la fois un ensemble de fonctions bien connues, et une interface abstraite pour des tâches qui dépendraient normalement du matériel et du système d'exploitation [18].

5.4 Codage

Comme expliqué auparavant notre solution passe par plusieurs étapes et chacune de ces étapes a une étape qui lui convient en programmation :

5.4.1 Récupération des données

Pour cette étape nous devons faire recours à des capteurs qu'on placera à l'intérieur de la voiture et qui se chargeront de récupérer toutes les valeurs dont nous avons besoin. À défaut de manque de matériels nous avons choisi d'introduire au code des méthodes pour générer ces valeurs d'une façon aléatoire. On aura donc deux méthodes aléatoires ; une qui génère des enchaînements de nœuds sous forme de listes en commençant par le premier nœud, Rentrer dans la voiture(Rv)

jusqu'au nœud démarrage (Dm). La deuxième méthode a pour mission d'attribuer des valeurs à ces nœuds, tout en respectant les intervalles cités auparavant dans la partie proposition.

5.4.2 Formation des graphes séquentiels

Les méthodes précédentes se chargent de fournir des successions de nœuds aléatoirement qui seront stockées sous forme de listes. Ensuite, afin de faciliter leurs compréhensions par l'entité machine, on a choisi de représenter ces listes par des matrices, qui font objet de graphes séquentiels. Ceci dit, à la fin de chaque essai une matrice est formée avec comme valeur « 1 » quand il existe une liaison entre le nœud A et le nœud B et la valeur « 0 » quand il y en a pas. Pour cette simulation on a fixé le nombre d'essais à 10. A la fin de cette étape nous obtiendrons 10 matrices avec les liaisons existantes.

5.4.3 Calcul de probabilités

Le calcul de probabilités se résume à une simple de division pour chaque paire de nœuds. On devra d'abord faire la somme des 10 matrices précédentes pour obtenir le nombre de fois qu'une liaison entre deux nœuds A et B a eu lieu. Ensuite nous allons parcourir les anciennes listes représentant les graphes séquentiels pour obtenir la valeur du nombre d'occurrence de chaque nœud dans les 10 essais. La division entre ces valeurs nous donnera des probabilités qu'on regroupera dans une seule matrice.

5.4.4 Calcul de temps

Puisque la notion du temps est liée à la vie réelle et pour manque de matériel, on a choisis de ne pas l'utiliser pour cette simulation. En effet donner des valeurs aléatoires représentant le temps ne pourrait que égarer l'exécution du processus. Néanmoins en cas d'obtention de matériel , nous avons défini une méthode de calcul du temps au niveau de la partie proposition.

5.4.5 Valeurs des nœuds

La méthode d'attribution des valeurs aléatoires de noeuds se charge d'associer a chaque liste de noeuds représentant un graphe séquentiel, une liste de valeurs. Chaque valeurs de cette liste rentre dans l'intervalle précisé correspondant a un noeud en particulier. Cette liste sera ensuite transformée en matrice tel que chaque noeuds du graphe lui correspond une ligne dans cette matrice, et que chaque poids de noeud lui correspond une case dans cette ligne en intersection avec le prochain noeud par rapport au graphe.

5.4.6 Calcul des poids de nœuds

Comme c'est déjà expliqué, chaque valeur d'action est considérée comme poids de nœud de cette action dans le graphe général, les nœuds aussi ont des poids qui est calculer comme suit :

-
- Pour chaque nœud A On parcourt tous les graphes séquentiels en comptant le nombre d'apparitions de A
 - On parcourt une deuxième fois tous les graphes séquentiels en additionnant les valeurs du nœud A : $V_1, V_2, V_3 \dots V_n$ (les poids du nœud A dans les graphes séquentiels)
 - Enfin, on calcule la moyenne entre ces valeurs en divisant sur le nombre d'apparition, le résultat est le poids du nœud A dans le graphe général

5.4.7 Graphe de référence

En programmation le graphe de référence sera représenté par deux matrices. En effet la première matrice est celle des probabilités de transition d'un nœud A vers un nœud B, la deuxième étant la matrice de poids de nœuds défini précédemment. La séparation des deux matrices est faite pour faciliter les calculs et pour faciliter la manipulation des valeurs, de ce raisonnement chaque valeur est facile d'accès et est facile de modification. Comparaison

Comme dans la partie proposition, l'étape de comparaison se divise en deux sous étapes ; la déduction d'un sous graphe similaire au nouveau graphe séquentiel généré après la phase d'apprentissage et la comparaison entre ces deux nœuds.

Déduction d'un sous graphe

Cette étape représente en programmation la sélection des cases concernées. En effet comme pour les graphes précédents, le nouveau graphe séquentiel aura deux matrices. L'enjeu est de trouver et de récupérer les valeurs des cases des matrices du graphe de référence similaires aux cases contenues dans les matrices du nouveau graphe. Les cases sélectionnées représenteront un sous graphe du graphe de référence comparable au nouveau graphe.

Comparaison

En programmation, cette étape suit le même raisonnement pour les deux sous comparaisons citées précédemment dans la partie proposition. En effet, il s'agit de comparer entre les valeurs des deux matrices probabilités et des deux matrices poids de nœuds des deux graphes à comparé. Le traitement est simple, une fois identifiées, les valeurs des cases concernées seront soustraites deux par deux et cela par rapport au nœud et à la transition ; deux valeurs du même nœud dans les deux graphes et deux valeurs de la même transition dans les deux graphes. Ces soustractions passeront par une division et une addition pour avoir deux valeurs de dissimilarité qu'on jugera grandes ou petites selon un certain seuil.

5.5 Conclusion

Dans ce chapitre, nous avons essayé de concrétiser notre proposition en commençant par décrire brièvement l'environnement de mise en œuvre de notre solution. Nous sommes ensuite passés à l'étape la plus importante dans le chapitre qui est l'explication détaillée des étapes suivies

pour la réalisation de notre système. En effet nous avons expliqué les étapes et les structures utilisées pour la numérisation des deux parties de notre solution ; la partie modélisation et la partie comparaison.

Conclusion générale

Chaque jour la technologie de l'information fait un autre pas en avant dans tous les domaines et continue de montrer son importance et son potentiel. Après avoir travaillé sur ce thème ; l'authentification biométrique du chauffeur approche comportementale ; nous pouvons dire que c'est et ça restera une solution la plus pertinente jusqu'à présent pour le problème de sécurité du véhicule.

Nous avons essayé d'ajouter notre touche à ce domaine en travaillant sur l'authentification approche comportementale, qui est une approche un peu mise de côté par les chercheurs des STI. Les voitures intelligentes peuvent être une solution à la situation réelle, le vol de voiture qui ne cesse d'augmenter au cours de ces dernières années partout dans le monde et la sécurité, qui était le sujet et le but de notre travail. Notre solution a été divisée en deux parties, chacune avec ses propres avantages. la première partie étant La modélisation des comportements du conducteur avec des graphes, cette dernière est une modélisation rarement utilisée. Notre solution permet à chaque chauffeur d'avoir un graphe propre à lui qui définit ses habitudes au sein de son véhicule. La deuxième partie est la comparaison entre le graphe générale du conducteur et le graphe généré par le système dès une utilisation du véhicule après la phase d'apprentissage. Ce type d'algorithme est peu utilisé en sécurité par rapport aux méthodes classiques ce qui lui offre l'avantage d'être plus robuste aux attaques.

Dans un future proche nous envisageons d'améliorer notre proposition et de l'optimiser afin qu'elle soit plus souple à l'utilisation quotidienne. Dans un premier lieu nous souhaitons prendre en compte plusieurs autres caractères propre à la façon de conduite du chauffeur, ceci pour rajouter de la précision à notre méthode. Au lieu de se contenter des actions faites avant le démarrage nous pousserons notre méthode à prendre en compte des actions après le démarrage. Ensuite, nous envisageons d'introduire une méthode pour vérifier la sécurité du chauffeur et non seulement la sécurité du conducteur. Cette méthode pourra sauver plusieurs vies humaines.

Bibliographie

- [1] PIER JEAN-PAUL, *Development Of Mathematics*, Springer Science Business Media, 2000.
- [2] FAGES JEAN-GUILLAUME, « Exploitation De Structures De Graphe En Programmation Par Contraintes », 2014.
- [3] Arifuzzaman S. et Khan M., « Fast Parallel Conversion of Edge List to Adjacency List for Large-Scale Graphs », *in : SpringSim* (2015).
- [4] Frank HARARY, « The Determinant Of The Adjacency Matrix Of A Graph », *in : SIAM Journal of computing(SICOMP)* 4 (1962), p. 202-211.
- [5] Gori MARCO et Sarti LORENZO, « Exact and Approximate Graph Matching Using Random Walks », *in : IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, (JULY 2005).
- [6] Emmanuel FERREIRA, « Apprentissage automatique en ligne pour un dialogue homme-machine situé », 2016.
- [7] Arthur SAMUEL, « Some Studies in Machine Learning Using the Game of Checkers », *in : IBM Journal of Research and Development* 3.3 (Juillet 1959), p. 210-229.
- [8] HAMMOUD Djamila épouse BOUDJAADA, « Apprentissage Automatique dans un Agent », 2011.
- [9] Chen SEI-WANG, Fang CHIUNG-YAO et Tien CHIH-TING, « Driving behaviour modelling system based on graph construction », *in : Elsevier* 26 (Janvier 2013), p. 314-330.
- [10] Roy ADITI, Halevi TZIPORA et Memon NASIR, « AN HMM-BASED BEHAVIOR MODELING APPROACH FOR CONTINUOUS MOBILE AUTHENTICATION », *in : IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (14 Juillet 2014), p. 3789-3793.
- [11] Lozoya-Santos Jorge de J. et al., « Survey on biometry for cognitive automotive systems », *in : Elsevier* 55 (Juillet 2019).
- [12] Banirostam HAMID, Shamsinezhad ELHAM et Banirostam TOURAJ, « Functional Control of Users by Biometric Behavior Features in Cloud Computing », *in : IEEE Computer Society* (Janvier 2013), p. 94-98.
- [13] Kalikova JANA et Krcal JAN, « Driver Identification Using Ear Biometrics, JOURNAL = IEEE, YEAR = 03 Janvier 2019, pages = 1277-1281, » *in : ()*.
- [14] Mondal SOUMIK et Bours PATRICK, « Continuous authentication using behavioural biometrics », *in : IEEE Computer society* 15 (Août 2013).

-
- [15] Yang CHING-HAN, Chang CHIN-CHUN et Liang DERON, « A Novel GMM-Based Behavioral Modeling Approach for Smartwatch-Based Driver Authentication », *in* : *MDPI sensors* 18 (28 mars 2018).
- [16] Yang CHING-HAN et al., « New Non-Intrusive Authentication Method based on Dynamics of Driver's Upper Body Joint Angles », *in* : *IEEE* 15 (16 Juillet 2015).
- [17] Augustyniak PIOTR et Slusarczyk GRA_zyna, « Graph-based representation of behavior in detection and prediction of daily living activities », *in* : *Elsevier* (2018), 261–270.
- [18] *Java*, Oracle, URL : https://java.com/fr/about/whatis_java.jsp.
- [19] URL : www.eclipse.org.

Résumé

Le domaine de sécurité est un domaine très actif en activité parallèle avec le domaine du piratage. En effet, le monde du piratage reçoit chaque jour des milliers de nouvelles techniques d'intrusions, de nouvelles méthodes pour le vol de données confidentielles et sensibles parfois en relation avec la vie humaine. Avec le développement croissant de ce dernier, il n'est plus envisageable d'utiliser les méthodes de sécurité classiques, il faut désormais penser à créer des systèmes intelligents robuste qui résisteront au maximum aux attaques des pirates. La problématique traitée dans ce travail concerne les STI et plus exactement l'authentification d'un chauffeur avec une approche comportementale. Le système proposé met le trait sur quelques comportements propre au chauffeur et fait recours à la théorie des graphes pour leur modélisation. La partie machine learning se charge du coté intelligent du système et est employé dans la prise de décision. Un tel système s'adapte aux changements et établie une vérification du conducteur à chaque entrée dans la voiture jusqu'au démarrage.

Abstract

The security field is a very active domain in parallel with the hacking domain. In fact, the world of hacking receives thousands of new intrusion techniques every day, new methods for the theft of confidential and sensitive data sometimes related to human life. With the growing development of this field, it is no longer possible to use traditional security methods, it is now necessary to think of creating robust intelligent systems that will resist as much as possible to attacks from hackers. The problem dealt with in this work concerns ITS and more exactly the authentication of a driver with a behavioural approach. The proposed system is based on some behaviours specific to the driver and makes use of the theory of graphs for their modelling. The machine learning part takes care of the intelligent side of the system and is used in decision-making. Such a system adapts to changes and establishes a driver check in each entry into the car until the start.