

TÉLÉCOM PARISTECH

PROJET DE FILIÈRE SR2I

Détection d'anomalies de classification dans l'IoT via Machine Learning

Antoine Urban, Yohan Chalier

encadré par
Jean-Philippe MONTEUUIS
Houda LABIOD

14 juin 2018

Résumé

Dans le contexte de l'internet des objets, les messages peuvent contenir de fausses informations générées par un utilisateur authentifié. Par conséquence, les mécanismes de sécurité reposant sur le chiffrement et la signature sont inutiles face à ces attaques. Les algorithmes reposant sur l'apprentissage machine (Machine Learning) permet de classer si un objet émettant des données est malicieux ou non

I Introduction

L'exercice de la conduite est synonyme de changements constant. Ainsi, lorsque nous conduisons, notre attention est tout entière consacré à notre environnement, car notre sécurité est celle des personnes qui nous entourent sont en jeu. Nous prêtons ainsi naturellement une attention toute particulière aux obstacles qui peuvent surgir, qu'il s'agisse tout simplement d'autres voitures partageant la route, des piétons ou encore des motocyclistes. Longtemps objets fantasmés, les véhicules autonomes deviennent aujourd'hui réalités grâce aux incroyables progrès réalisés dans les domaines de l'intelligence artificielle et du développement de capteurs. Si ces processus visent à remplacer le conducteur humain, il semble naturel qu'ils prêtent la même attention aux obstacles. Les méthodes utilisées se doivent d'être fiables et performante, car en matière de sécurité, le droit à l'erreur n'existe pas !

Dans ce travail, nous nous concentrerons sur les 3 types d'obstacles les plus communs sur une route :

- Une voiture ;
- Une moto ;
- Un piéton.

Tout travail sur l'analyse d'obstacles nécessite la détection préalable de régions d'intérêt en utilisant par exemple la méthode du "point-in-polygon" [1], ou encore à l'aide de capteurs à ultrasons et d'analyse de signal[?]. Ce travail n'a pas comme objectif de revenir sur l'utilisation de ces techniques, mais se concentrera sur l'analyse à posteriori des dimensions ainsi collectées avec pour but de proposer une classification efficace.

Objectifs

A travers un cas simple, nous cherchons à vérifier l'appartenance d'un objet à une classe (e.g. un véhicule) en utilisant un algorithme de machine learning et les dimensions de l'objet. Il s'agit donc de proposer un modèle de classification multi-classes en réalisant un classeur à partir d'un algorithme d'apprentissage supervisé. Au préalable, nous nous assurons de trouver des bases de données avec les dimensions pour chacune des classes pour entraîner et valider notre modèle. Ensuite, nous proposerons une méthodologie de sélection d'un algorithme d'apprentissage supervisé en évaluant leur précision et leur efficacité respectives. Dans ce travail, nous avons considérés les algorithmes (c) suivant :

- Réseau de Neurones ($c = 1$) ;
- Adaboost ($c = 2$) ;
- SVM($c=3$) ;
- Random Forest ($c=4$).

La fonction suivante permet de résumer le déroulé du processus de prédiction :

Algorithm 1: Fonction de production

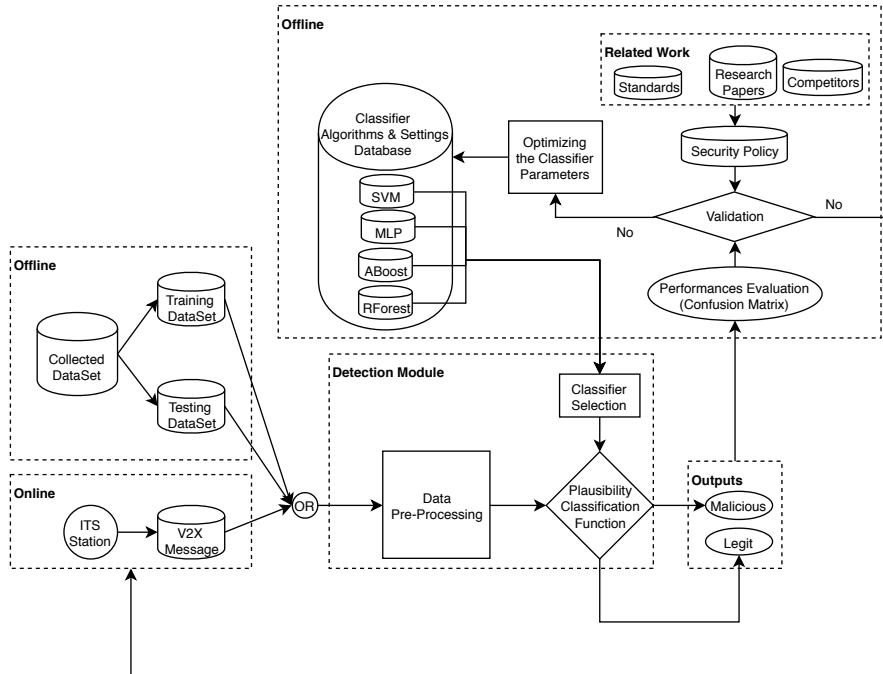
Donnée : Un message i envoyé par un objet communicant contenant :
 — Classe de l'objet ($Classe_i = (Voiture, moto, piéton)$)
 — Dimension de l'objet (Longueur : L_i ; Largeur : l_i).

Result: Détection des objets malicieux

```

if function (Class, Longueur, Largeur) then
  | Return “malicieux”;
else
  | Return “non malicieux”;
end

```



II Méthode d'évaluation

II.1 Matrice de confusion

Une technique pour évaluer la performance d'un algorithme de classification est d'utiliser une matrice de confusion. Il s'agit d'un résumé des résultats de prédiction sur un problème de classification, qui donne un aperçu du degré de confusion de notre modèle.

Terminologie

Pour définir de manière formelle une matrice de confusion, il convient d'introduire les termes suivant :

- TP (Vrai Positif) : item correctement détecté positif
- FP (Faux Positif) : item déclaré positif, là où il est en réalité négatif.
- FN (Faux négatif) : item déclaré négatif alors qu'il était en réalité positif.
- TN (Vrai négatif) : item correctement détecté comme négatif

Ces derniers permettent de définir la classification, correct ou non, des objets données en entrée de notre fonction. Nous pouvons dès lors construire la matrice suivante :

		Actual class			
		Positive	Negative		
Predicted classes	Positive	TP	FP	PPV	FDR
	Negative	FN	TN	FOR	NPV
		TPR	FPR		
		FNR	TNR		

Pour la suite, et pour comprendre cette matrice, il convient d'introduire les résultats suivants :

- Ratio de Vrai positif : $TPR = \frac{TP}{TP + FN}$
- Ratio de Vrai négatif : $TNR = \frac{TN}{TN + FP}$
- Ratio de Vrai positif : $TPR = \frac{TP}{TP + FN}$
- Ratio de Vrai négatif : $TNR = \frac{TN}{TN + FP}$
- Ratio de Faux positif : $FPR = \frac{FP}{TP + TN}$
- Ratio de Faux négatif : $FNR = \frac{FN}{TP + FN}$
- Valeur prédictive positive : $PPV = \frac{TP}{TP + FP}$
- Valeur prédictive négative : $NPV = \frac{TN}{TN + FN}$
- Taux de fausses découvertes : $FPR = \frac{FP}{TP + FP}$
- Taux de fausses omissions : $FOR = \frac{FN}{TN + FN}$

Ce mode de représentation permet non seulement de mettre en évidence les erreurs qui sont faites par votre classificateur, mais surtout des types d'erreurs

qui sont commises. En effet, l'utilisation de la seule précision peut être trompeuse dans la mesure où le nombre d'observation dans chaque classe est inégale. Pour illustrer ce phénomène, prenons un exemple précis. Considérons l'algorithme SVM (support vector Machine) avec les paramètres suivants :

C=0.1, cache-size=200, class-weight=None, coef0=0.0, decision-function-shape='ovr', degree=3, gamma='auto', kernel='linear', max-iter=-1, probability=False, randomState=None, shrinking=True, tol=1e-05, verbose=False

La précision de cet algorithme avec ces paramètres est égale à 0.96660 et renvoie la matrice de confusion suivante :

$$\begin{bmatrix} 2771 & 74 \\ 239 & 239 \end{bmatrix}$$

Nous remarquons bien que la classification est loin d'être satisfaisante alors que le score de précision est, lui, important. Il y a en effet dans ce dernier aucune pondération. Si une classe contenant un nombre de paramètres important est bien classifiée, elle peut "masquer" les mauvais résultats obtenus pour d'autres classes plus petites.

Bibliographie

- [1] Deepika, N AND Sajith Variyar, V. V., Obstacle classification and detection for vision based navigation for autonomous driving, Addison Wesley, Massachusetts, IEEEI, 2017.
- [2] Gerard Gibbs and Huamin Jia and Irfan Madani, Obstacle Detection with Ultrasonic Sensors and Signal Analysis Metrics, Transportation Research Procedia, 2017.