Une méthodologie pour la Sélection de Variables pour la Stéganalyse

A Feature Selection Methodology for Steganalysis

Yoan Miche [1,2], Patrick Bas [1,2], Amaury Lendasse [1], Christian Jutten [2], Olli Simula [1]

[1] Helsinki University of Technology – Laboratory of Computer and Information Science P.O. Box 5400, FI-02015 HUT, Finland
[2] GIPSA-lab/Images and Signal Department CNRS, INPG, UJF, Grenoble, France

Manuscrit reçu le 15 avril 2008

Résumé et mots clés

Le principe de la stéganalyse est de classer un document incriminé comme original ou comme stéganographié. Cet article propose une méthodologie pour la stéganalyse utilisant la sélection de caractéristiques, orientée vers une diminution des intervales de confiance des résultats habituellement donnés. La sélection de caractéristiques permet également d'envisager une interprétation des caractéristiques d'images sélectionnées, dans le but de comprendre le fonctionnement intrinsèque des algorithmes de stéganographie. Il est montré que l'écart type des résultats obtenus habituellement en classification peut être très important (jusqu'à 5 %) lorsque des ensembles d'entrainements comportant trop peu d'échantillons sont utilisés. Ces tests sont menés sur six algorithmes de stéganographie, utilisés avec quatre taux d'insertions différents : 5, 10, 15 et 20 %. D'autre part, les caractéristiques sélectionnées (généralement 10 à 13 fois moins nombreuses que dans l'ensemble complet) permettent effectivement de faire ressortir les faiblesses ainsi que les avantages des algorithmes utilisés.

Stéganographie, sténaganalyse, sélection de variables, méthodologie

Abstract and key words

Steganography has been known and used for a very long time, as a way to exchange information in an unnoticeable manner between parties, by embedding it in another, apparently innocuous, document. Nowadays steganographic techniques are mostly used on digital content. The online newspaper Wired News, reported in one of its articles [2] on steganography that several steganographic contents have been found on web sites with very large image database such as eBay. Niels Provos [3] has somewhat refuted these ideas by analyzing and classifying two million images from eBay and one million from USENet network and not finding any steganographic content embedded in these images. This could be due to many reasons, such as very low payloads, making the steganographic images very robust and secure to steganalysis.

The security of a steganographic scheme has been defined theoretically by Cachin in [1] but this definition is very seldomly usable in practice. It requires to evaluate distributions and measure the Kullback-Leibler divergence between them.

In practice, steganalysis is used as a way to evaluate the security of a steganographic scheme empirically: it aims at detecting whether a medium has been tampered with – but not to detect what is in the medium or how it has been embedded. By the use of features, one can get some relevant characteristics of the considered medium, and assess, by the use of machine learning tools, usually, whether the medium is genuine or not. This is only one way to perform steganalysis, but it remains the most common.

One of the main issues with this scheme is that people tend to use more and more features extracted from the media (we consider only JPEG images in this article) in order to increase the performances of detection of modified images. This number of features corresponds to the dimensionality of the space in which are performed machine learning processes (typically, training of a classifier). This usually leads to very high dimensional spaces for which many problems arise (in comparison to low dimensional spaces): mainly, the required number of images to have an appropriate filling of the space in which the classifier is trained, is never reached. This filling is required for the classifier to train on properly distributed data among the feature space. Also, when the number of features is too high, interpretation of the most relevant features becomes very difficult if not to say impossible.
In this article, some of the problems encountered because of the high dimensionality of the problem usually met in steganalysis, are presented, along with possible solutions.
To the problem of the required number of images for filling the space, is proposed an evaluation of a sufficient number of images: a bootstrap algorithm is used to estimate the variance of the classifier's results for different amounts of images. Once the variance is low enough to have accurate results, the number of images required for that number of features is attained.
With this sufficient number of images, feature selection is then performed, with a forward algorithm, in an attempt to decrease the dimensionality and also to gain interpretability over which features have been reacting the most. Hence, a knowledge of the steganographic's scheme can be inferred and its scheme could be modified accordingly to improve its security.
These ideas are combined in a methodology, which is tested on 6 different steganographic algorithms, for different sizes of the embedded information. The result is an estimation of the sufficient number of images for obtaining results with low enough variance. Selected sets of features also enable to keep the same performances (within the small variance range) while providing insights on the weaknesses of each algorithm. These weaknesses are analyzed separately for each algorithm.
In conclusion, the proposed methodology enabled to estimate the variance of typically given results for steganalysis, along with added interpretability. The proposed reduced sets of features have also made it possible to keep the same performances as for the full set.

References

[1] C. CACHIN, An information-theoretic model for steganography. In *Information Hiding : 2nd international Workshop*, volume 1525 of *Lecture Notes in Computer Science*, pages 306-318, 14-17 April 1998.
[2] D. MCCULLAGH, Secret Messages Come in. Wavs. online Newspaper : Wired News, February 2001. http://www.wired.com-/news/politics/0,1283,41861,00.html.
[3] N. PROVOS, Defending against statistical steganalysis. In *10th USENIX Security Symposium*, pages 323-335, 13-17 April 2001.

1. Introduction

La stéganographie est utilisée depuis l'antiquité sous diverses formes, comme un moyen secret de transmission d'information : l'information à faire transiter – appelée stéganogramme – est dissimulée au sein d'un document ou objet de façon à ce qu'une personne non informée de la présence du stéganogramme ne puisse identifier une quelconque « modification » du document support. Un exemple type de stéganographie consiste à dissimuler des bits d'information dans une image, par exemple. L'image en question devra donc présenter le minimum de distorsions visuelles et statistiques pour qu'aucune entité ne puisse identifier l'image comme suspecte, et donc que l'information dissimulée parvienne au seul destinataire sans encombre.

Actuellement, la plupart des techniques stéganographiques utilisent des supports numériques : images, musique, vidéos... Le journal Wired News relate dans un article à propos de la stéganographie [12], qu'un grand nombre de contenus stéganographiés a été trouvé sur des sites comportant une base d'images très conséquente, par exemple le site eBay. Ces faits ont été modérés par une analyse menée par Niels Provos [18], utilisant deux millions d'images obtenues depuis eBay et le réseau USENet, pour lesquelles aucun contenu stéganographique n'a pu être détecté.

Ceci peut être dû à de très faibles quantités d'informations insérées dans les images analysées. En effet, si la quantité d'information dissimulée dans le contenu support est faible, les distorsions dues à la stéganographie restent également faibles, ce qui permet d'augmenter la sécurité d'une certaine manière, rendant la détection inefficace.

Ce concept de sécurité pour la stéganographie reste difficile à définir et à évaluer en pratique. Une version théorique proposée par Cachin dans [2], utilise la divergence de Kullback-Leibler et définit la sécurité d'un procédé stéganographique comme suit :

Le procédé est déclaré ε-sûr si la divergence de Kullback-Leibler δ entre la distribution p_{original} du medium support (en supposant que ce medium support a donc été généré par cette distribution) et celle p_{stego} du medium qui contient l'information cachée, est plus faible que ε :

$$\delta(p_{\text{original}}, p_{\text{stego}}) \leq \varepsilon. \qquad (1)$$

Pour le cas où $\varepsilon = 0$, le procédé stéganographique considéré est déclaré *sûr*. La stéganographie est alors considérée comme parfaite puisqu'aucune différence statistique n'est créée, lorsqu'une information est embarquée dans le medium support – une image, pour le cadre de cet article. Dans ce cas de sûreté maximum la stéganalyse est théoriquement impossible, les deux versions du medium étant d'un point de vue statistique, totalement identiques.

De telles performances pour un algorithme de stéganographie restent toutefois impossibles à atteindre à l'heure actuelle. D'autre part, plus la quantité d'information à dissimuler dans le medium est importante, plus les distorsions, d'ordres statistiques ou visuelles, sont importantes ; la sécurité est donc plus faible. Afin de mesurer la quantité d'information dissimulée dans un medium, on utilise le *taux d'insertion*, qui se définit comme suit :

Soit une méthode de stéganographie S et un medium M ; la méthode de stéganographie annonce qu'elle peut dissimuler au maximum T_{Max} bits d'information dans M (cette quantité dépend largement de la méthode de stéganographie ainsi que du medium et se nomme *capacité d'insertion*), de par son fonctionnement. Le taux d'insertion T est définit comme la part de T_{Max} effectivement utilisée par l'information à dissimuler.

Ainsi pour une information de taille T_i bits, le taux d'insertion T est $T = T_i / T_{\text{Max}}$ (la plupart du temps exprimé en pourcentage).

D'autres quantités permettent d'obtenir un rapport entre la taille de l'information à dissimuler et les caractéristiques de la méthode stéganographique et du medium (nombre de bits par coefficient DCT (Direct Cosine Transform) non nul, etc). Toutefois, le taux d'insertion a l'avantage de se baser sur l'estimation de la méthode de stéganographie directement (en terme de capacité d'insertion) et donc de permettre une comparaison plus « équitable » entre les différentes méthodes.

Puisque la possibilité d'une évaluation théorique de la sécurité d'une méthode de stéganographie pour différents taux d'insertion n'est pas réalisable (l'estimation des distributions permettant la génération des media originaux et stéganographiés s'avère très complexe), la voie empirique est utilisée, au travers de la stéganalyse.

De la même facon que la cryptanalyse vise à « casser » les méthodes de cryptographie, la stéganalyse cherche à identifier les media qui ont été modifiés par stéganographie et sont susceptibles de contenir une information cachée. Il ne sagit pas de révéler le contenu dissimulé, mais bien d'identifier un medium comme « suspect » avec une certaine probabilité associée.

Pour le cadre de cet article, les images de type JPEG sont utilisées comme media servant à dissimuler de l'information – pour la suite de cet article, les images JPEG seront les seuls media considérés pour la stéganographie. Ceci pour la simple raison que les images JPEG sont parmi les plus répandues dans le monde numérique et qu'un nombre conséquent d'algorithmes de stéganographie existe pour ce type d'images. Egalement, de par les spécificités de l'algorithme de compression JPEG [3] (qui ne seront pas détaillées ici), la dissimulation d'information peut se faire d'un certain nombre de manières, qui peuvent dans certaines mesures empêcher des détections statistiques classiques (histogrammes de coefficients DCT notamment).

Parmi les techniques récentes de stéganalyse, on trouve la « stéganalyse par caractéristiques ». L'idée de ce procédé est d'extraire des caractéristiques diverses d'une image JPEG, telles que les différents histogrammes des coefficients DCT, des mesures de variations entre pixels au niveau des raccords entre blocs JPEG (typiquement 8×8), etc. Globalement, on observe une augmentation particulièrement forte du nombre de caractéristiques utilisées pour la stéganalyse. Les premiers exemples de

stéganalyse basée sur les LSB (Least Significant Bits, bits de poids faible) utilisaient un nombre de caractéristiques inférieur à la dizaine et propre à chaque algorithme de stéganographie afin de détecter la possible présence d'une information dissimulée. En 1999, Westfeld propose un modèle d'attaque statistique basé sur un test du χ^2 sur les LSB des coefficients DCT des images analysées [25]. Plus récemment, en 2004, Fridrich dans [5] utilise un ensemble de 23 caractéristiques, tandis que Farid *et al.* proposaient déjà un ensemble de 72 caractéristiques en 2002 [11]. Depuis ces travaux, un nombre croissant de recherches fait usage de nombres de caractéristiques particulièrement élevé. On peut citer récemment Y. Q. Shi *et al.* [21], qui proposent un ensemble de 324 caractéristiques, basées sur des différences de blocs JPEG modélisées par des processus markoviens.

L'idée principale derrière cette augmentation était celle de « stéganalyseur universel ». En effet, les premiers modèles de stéganalyse permettaient d'identifier une seule méthode de stéganalyse à la fois. Il fallait donc utiliser chacun de ces modèles séparément sur une image à analyser, afin de vérifier la possible présence d'une information. L'aspect universel des nouvelles approches souhaite détecter l'ensemble des méthodes de stéganographie, quelles qu'elles soient. À nouveau, le but direct n'est pas de pouvoir dire quelle méthode de stéganographie a été utilisée, mais bien d'identifier l'image comme suspecte ou non (même si la recherche de la méthode de stéganographie utilisée est une suite logique à ce concept de stéganalyseur universel).

Bien que l'ajout de ces caractéristiques permette en effet d'améliorer les performances globales de stéganalyse, on ne peut s'empêcher de noter certains effets néfastes dus à cette augmentation : typiquement, la stéganalyse utilise l'ensemble des caractéristiques extraites d'une image directement avec un classifieur à apprentissage supervisé, de type Support Vector Machine (SVM), en général. Ce classifieur entraîné au préalable, permet d'obtenir une probabilité concernant la possibilité que l'image ait été modifiée par stéganographie ou non. Un des problèmes étudié dans cet article est lié directement à ce concept de stéganalyseur universel et au nombre de caractéristiques extraites de l'image.

En effet, le nombre de caractéristiques utilisé définit directement la dimension de l'espace dans lequel le classifieur utilisé doit établir une frontière entre images originales et stéganographiées (pour reprendre l'idée de frontière du SVM). L'entraînement du classifieur se fait donc dans un espace à haute dimensionalité pour un stéganalyseur universel. Or il a été montré, par exemple dans [4, 23] que la dimensionalité de l'espace dans lequel est entraîné le classifieur peut avoir une incidence cruciale sur ses performances, relativement au nombre d'images utilisées. Ainsi, un nombre d'images insuffisant au regard de la dimensionalité de l'espace peut entraîner un mauvais apprentissage du classifieur et donc des résultats faussés ou ayant une importante variance statistique.

La section suivante détaille un certain nombre de problèmes liés à la dimensionalité des données et fréquemment rencontrés dans le domaine de la stéganalyse : le phénomène d'espace vide et de concentration des distances en grande dimension, l'importante variance des résultats lorsque le nombre d'images est insuffisant ainsi que le manque d'interprétabilité des résultats lors de l'utilisation d'un tel nombre de caractéristiques. Une méthodologie (dont le principe est illustré Figure 1) permettant de résoudre ces différents problèmes est ensuite proposée, par la détermination d'un nombre suffisant d'images pour un entraînement « fiable » (en termes de variance des résultats de classification) du classifieur utilisé, puis par une sélection de caractéristiques donnant une interprétabilité accrue aux résultats du classifieur.

Cette méthodologie est testée en section 4 sur six méthodes de stéganographie différentes et pour quatre taux d'insertion différents. Les résultats en classification sont ensuite interprétés grâce à aux caractéristiques sélectionnées par la méthodologie comme les plus pertinentes pour chaque méthode de stéganographie.

2. Problèmes liés à la dimensionalité

L'expression commune « Malédiction de la dimensionalité » [1] se rapporte à l'ensemble des problèmes rencontrés lors de l'utilisation de données possédant une grande dimensionalité. Dans cette section, certaines conséquences de cette dimensionalité sont étudiées, relativement au nombre d'images et au nombre de caractéristiques.

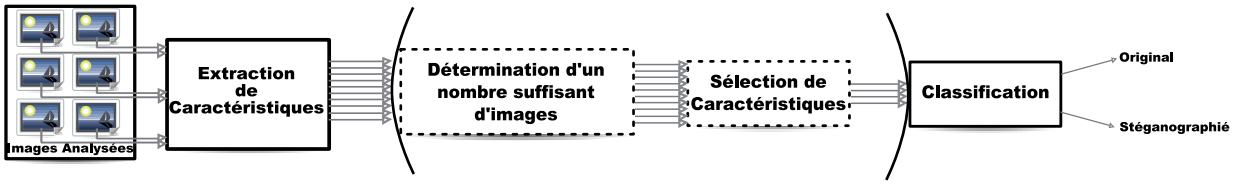


Figure 1. Schéma de la méthode classique de stéganalyse pour une image : les caractéristiques sont tout d'abord extraites de l'image considérée et utilisées par un classifieur (entrainé au préalable), permettant de décider si cette image est originale ou non. La méthodologie proposée ajoute deux étapes supplémentaires dont le but est de réduire la dimensionalité de l'ensemble de caractéristiques, grâce à la sélection de caractéristiques et d'obtenir des résultats fiables par la détermination d'un nombre suffisant d'images pour le problème.

2.1. Problèmes liés au nombre d'images

2.1.1. Le besoin d'images

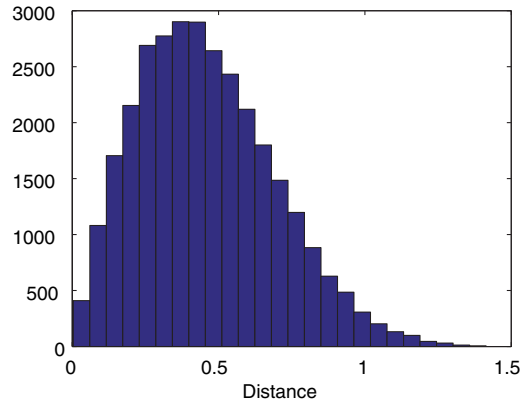
Pour illustrer ce problème dans le cas d'espaces de faible dimension, on peut imaginer quatre points (correspondant à quatre images placés aléatoirement dans un espace à deux dimensions (on aurait donc extrait deux caractéristiques de chacune des quatres images, qui sont les coordonnées dans l'espace) : il est impossible de déduire à partir de ces quatre points, une structure adéquate permettant de décrire leur répartition, et donc encore moins, l'élaboration d'un modèle permettant d'interpoler la position de ces points dans l'espace afin de prédire où se situeront les autres points suivant une distribution identique. En revanche, si l'on dispose de milliers de points, il est intuitivement beaucoup plus faisable d'élaborer un tel modèle.
De façon plus générale que pour ces cas à deux ou trois dimensions, le nombre de points (et donc d'images ici) nécessaire à la construction d'un modèle approprié grandit exponentiellement avec le nombre de dimensions utilisé (nombre de caractéristiques ici). En effet, pour le cas général d'un espace qui serait un hyper-cube en dimension d, la grille cartésienne de pas ε nécessite un nombre de points en $O((1/\varepsilon)^d)$, pour être correctement remplie (c'est-à-dire un point sur chaque intersection de la grille). Pour le cas où l'on aurait $d = 10$ (dix caractéristiques, donc) et une grille de pas 1/10, il faut donc 10^{10} points pour remplir la grille, et donc autant d'images pour avoir un espace correctement rempli pour que le modèle construit, le classifieur pour la stéganalyse, soit fiable et ne fasse que de l'interpolation. Pour le cas de la stéganalyse, les ensembles de caractéristiques les plus petits font déjà usage d'au moins 0 à 20 caractéristiques, portant le nombre de points théoriquement nécessaire à un remplissage correct de l'espace, au delà des limites atteignables (en termes de stockage et de traitement des données) : stocker 10^{10} images n'est pas trivial. Le problème d'extrapolation des modèles de classifieurs construits est donc en général très présent.

2.1.2. Le phénomène d'espace vide et de concentration des distances

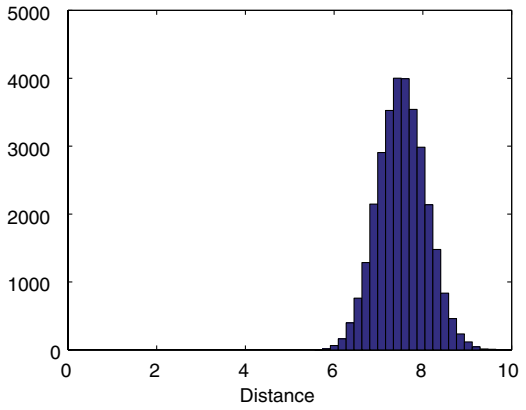
Ce phénomène peut être illustré par l'exemple suivant : d'un point de vue théorique, si l'on tire des points suivant une loi normale (moyenne nulle et variance unitaire, pour les calculs), en dimension d, et que l'on considère la probabilité d'avoir un point à la distance r de la moyenne (donc de zéro), la densité de cette probabilité est donnée par

$$f(r,d) = \frac{r^{d-1}}{2^{d/2-1}} \cdot \frac{e^{-r^2/2}}{\Gamma(d/2)} \qquad (2)$$

qui possède un maximum pour $r = \sqrt{d-1}$. Ceci implique que lorsque la dimensionalité augmente, les points ont tendance à s'éloigner de la moyenne de la distribution. Une conséquence directe de ce phénomène est que le centre de l'hypercube en dimension d va avoir tendance à être « vide », puisque les points vont se concentrer en majorité aux bords de l'espace considéré, c'est-à-dire dans les coins du cube, ici.
Un phénomène relié à celui de l'espace vide est celui de la concentration des distances. En effet, avec l'augmentation de la dimensionalité, les distances point à point ont tendance à se concentrer dans une plage de valeurs réduite, comme le montre la Figure 2.



(a) Dimension 2, loi uniforme



(b) Dimension 100, loi uniforme

Figure 2. Aperçu des distances point à point pour une loi uniforme en dimensions 2 et 100. En dimension 2, les distances varient sur toute la plage disponible (de 0 à $\sqrt{2}$), tandis qu'en dimension 100, les distances s'éloignent de zéro et se concentrent vers les plus grandes valeurs possibles.

Ainsi, lorsque la dimensionalité de l'espace augmente, les distances entre points ont tendance à être toutes les mêmes et à être importantes. Le nombre de points nécessaire au remplissage de l'espace de façon suffisante pour un apprentissage correct d'un classifieur est donc d'autant plus important : puisque les points se retrouvent très éloignés les uns des autres et sur les bords de l'espace considéré, il en faut d'autant plus, si l'on souhaite remplir le « centre » de l'espace et éviter d'entraîner un classifieur dans un espace particulièrement hétérogène en terme de « densité de points ».

2.1.3. L'augmentation de la variance des résultats

Enfin, une dernière raison motivant la détermination d'un nombre d'images suffisant à la construction d'un classifieur « fiable », est la variance des résultats. Seuls des arguments expérimentaux sont avancés pour ce problème : il a été observé que la variance des résultat d'un classifieur est importante lorsque le nombre d'images est faible par rapport aux nombre de caractéristiques extraites. En revanche, lorsque le nombre d'images est conséquent, la variance des résultats devient plus acceptable. Ce phénomène a été vérifié expérimentalement dans la section suivante.
Au final, les trois problèmes soulevés précédemment mènent à la même conclusion : il faut un nombre important d'images au vu du nombre de caractéristiques extraites. Puisque le nombre d'images que requiert la théorie (exponentiel par rapport au nombre de caractéristiques) est inaccessible, l'idée de la première étape de la méthodologie proposée, est de trouver un nombre « suffisant » d'images pour le nombre de caractéristiques extraites, au regard d'une valeur acceptable de la variance des résultats.

2.1.4. Solution proposée au problème du nombre d'images

Afin de déterminer un nombre suffisant d'images pour le problème d'entraînement du classifieur, une méthode empirique utilisant le bootstrap [8] est utilisée : il sagit de faire varier la taille de l'ensemble d'images utilisé pour l'entraînement et d'étudier la variance des résultats obtenus pour chaque taille d'ensemble.

2.2. Le manque d'interprétabilité des résultats

L'interprétabilité des résultats (et notamment des caractéristiques sélectionnées) ainsi que ses possibles applications, telle que l'analyse du fonctionnement de l'algorithme, de par les caractéristiques les plus sensibles pour la détection, est un autre argument pour la réduction de dimensionalité par la sélection de caractéristiques ; l'ensemble complet des 193 caractéristiques utilisé pour cet article permet en effet d'atteindre des performances importantes en classification, mais rend quasiment impossible une analyse des caractéristiques les plus utiles pour le problème. L'ensemble est en ce sens « trop » exhaustif.

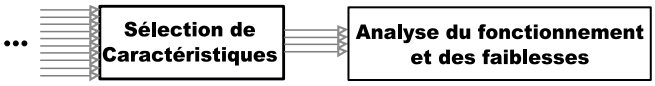


Figure 3. Illustration de l'idée d'analyse de l'algorithme, par sélection et mise en évidence des caractéristiques les plus sensibles.

Ainsi, la réduction du nombre de caractéristiques à un nombre très réduit, tout en conservant des performances similaires, permet de mieux comprendre et analyser les faiblesses et sensibilités d'un algorithme de stéganographie. Une telle analyse est proposée en section 4 pour les six procédés stéganographiques dans le cadre des expériences menées.
Au travers de l'analyse des caractéristiques séléctionnées, l'analyse de l'algorithme et de ses faiblesses à proprement parler, devient possible, comme présenté en Figure 3. De par cet analyse, il peut être possible d'identifier l'algorithme utilisé, si inconnu ou bien de mettre en évidence ses faiblesses, dans le but de comprendre son fonctionnement ou encore d'améliorer sa sécurité.

2.2.1. Solution proposée au problème d'interprétabilité

De manière à déterminer quelles caractéristiques, parmi les 193 utilisées, sont les plus importantes au vu des performances de classification, une sélection de caractéristiques est utilisée. La réduction du nombre de dimensions de l'espace d'entraînement du classifieur par cette sélection a également un effet bénéfique sur les problèmes liés à la dimensionalité et au nombre de points mentionnés précédemment.
La section suivante met en place les deux solutions aux problèmes développés précédemment, au travers d'une méthodologie en deux étapes.

3. Méthodologie de test d'un algorithme de stéganographie

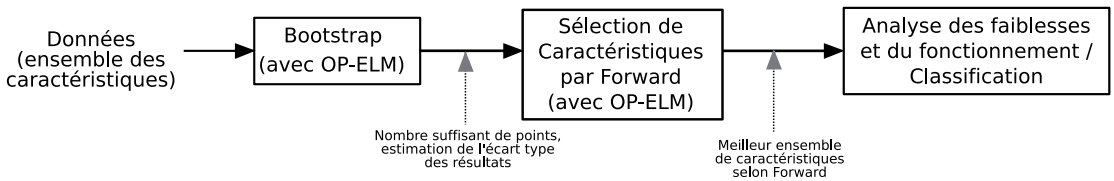


Figure 4. Schéma de la méthodologie proposée : (1) Un nombre suffisant de points au vu de la dimensionalité est déterminé par l'utilisation d'une méthode de bootstrap (pour obtenir une certaine stabilité statistique) ; (2) La sélection de caractéristiques par l'algorithme Forward est utilisée avec un classifieur OP-ELM. L'ensemble de caractéristiques obtenu est alors utilisé soit pour une stéganalyse classique (classification, ici), ou bien une analyse de l'algorithme de stéganographie.

3.1. Présentation du classifieur utilisé et notions de classification

3.1.1. Notions de classification

L'idée d'un classifieur est de construire un modèle permettant de discriminer différentes « classes » au sein de données. Un classifieur à apprentissage supervisé est utilisé dans cet article, c'est-à-dire que l'on fournit des données comprenant la classe de chaque point au classifieur, pour son apprentissage. Il existe des classifieurs à apprentissage non-supervisé. Un classifieur s'utilise en deux étapes : d'abord, il sagit de construire le modèle (le classifieur à proprement parler), c'est l'étape d'entraînement/validation. Une fois le modèle établit sur un ensemble de données (l'ensemble d'entraînement), on peut tester le classifieur sur des autres données (ensembles de test).
On dénote par k-fold cross-validation (ou k-fold) le fait de diviser l'ensemble de données servant à l'entraînement du classifieur en k sous-ensembles, dont $k-1$ vont servir à l'entraînement à proprement parler, et le sous-ensemble restant servira à valider le modèle entraîné, en calculant ses performances sur ce sous-ensemble. Ce processus est ensuite répété k fois, en utilisant une seule fois chacun des k sous-ensembles pour la validation et les autres $k-1$ pour l'entraînement.
Le Leave-One-Out est une version de la k-fold pour laquelle k est égal à la taille des données, c'est-à-dire que chaque point de l'ensemble de données va être utilisé seul une fois pour l'évaluation du modèle, tandis que le reste des données va servir à l'entraînement. Il a été montré [8] que ce type de k-fold cross-validation permet de s'affranchir de résultats de test, du fait de la fiabilité des résultats obtenus.

3.1.2. Classifieur utilisé : OP-ELM

Le classifieur OP-ELM (Optimally-Pruned Extreme Learning Machine) [13,22] est une variante de l'Extreme Learning Machine original de Huang [7]. Ce classifieur utilise des réseaux de neurones à couches multiples dont les coefficients sont initialisés aléatoirement. L'OP-ELM ajoute au modèle original d'autres noyaux ainsi qu'une étape permettant de supprimer les neurones les moins utiles du réseau. Les deux théorèmes sur lesquels l'ELM de Huang est basé ne seront pas détaillés ici mais peuvent être trouvés dans [7]. L'idée principale est d'utiliser un réseau de neurones avec une seule couche cachée (Single Hidden Layer Feedforward Neural Network), pour lequel les coefficients et les biais sont initialisés aléatoirement. La Figure 5 présente un modèle simplifié de réseau de neurones.
Au final, si le réseau de neurones modèle parfaitement la sortie $\mathbf{Y} = (y_1, \ldots, y_N)$, le réseau se résume à l'équation suivante :

$$\sum_{i=1}^{M} \beta_i f(\mathbf{w}_i \mathbf{x}_j + b_i) = \mathbf{y}_j, j \in [\![1, N]\!], \tag{3}$$

en notant N le nombre d'entrées $\mathbf{X} = (x_1, \ldots, x_N)$ et M le nombre de neurones dans la couche cachée.
Les coefficients $\mathbf{W}$ et $\mathbf{B}$ (biais) sont initialisés aléatoirement par l'ELM. Les nouveautés apportées par l'OP-ELM sont une plus grande robustesse de l'ELM original face à des données ayant certaines dimensions fortement corrélées, ainsi que l'utilisation d'autres fonctions f permettant d'utiliser OP-ELM pour des cas où le modèle à approximer possède des composantes linéaires, par exemple.
L'étape de validation de ce classifieur est effectuée par un Leave-One-Out, bien plus précis qu'une k-fold, et ne nécessite pas de test [8]. Il a été montré sur un certain nombre d'applications expérimentales [13,22], que l'OP-ELM donne des résultats très proches de ceux d'une SVM (Support Vector Machine) et se comporte d'une façon similaire, tout en ayant l'avantage de temps d'exécutions 10 à 100 fois plus faibles.

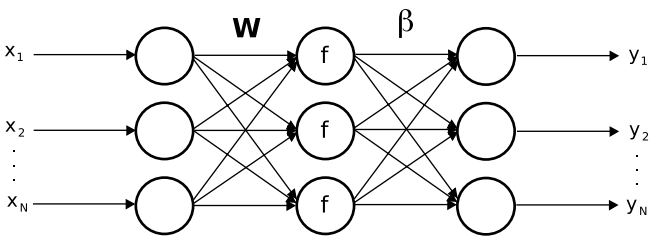


Figure 5. Schéma d'un réseau de neurones possédant une seule couche cachée et ne possédant pas de boucle (récursivité). Les valeurs d'entrée $\mathbf{X} = (x_1, \ldots, x_N)$ sont pondérées par les coefficients $\mathbf{W}$. Un biais $\mathbf{B}$ peut être ajouté (non représenté ici) et le résultat passe par une fonction d'activation f, laquelle est enfin pondérée par les coefficients de sortie β pour obtenir la sortie $\mathbf{Y} = (y_1, \ldots, y_N)$.

3.2. Détermination d'un nombre suffisant de points : Boostrap

Le nombre de points approprié pour la dimensionalité doit être déterminé, ou du moins, une estimation d'un nombre suffisant permettant un entraînement fiable du classifieur et des résultats de faible variance (on parlera plutôt de l'écart type des résultats dans la suite). Cette étape permet également de donner une valeur d'écart type des futurs résultats et rendre leur interpétation plus précise.
À cette fin, un classifieur OP-ELM est utilisé avec un algorithme de Bootstrap [8] sur 100 itérations : un sous-ensemble de l'ensemble complet des données est tiré aléatoirement avec possibles répétitions, et le classifieur est entrainé sur ces données. Lorsque suffisamment d'itérations sont utilisées pour le Bootstrap, une estimation fiable de l'écart type des résultats est obtenue.
Ce processus est répété pour des tailles croissantes du sous-ensemble tiré aléatoirement. La taille la plus grande de sous-ensemble utilisé est limitée par les temps de calculs, puisque le processus de Bootstrap doit être itéré autant que possible (100 itérations ici). L'écart type des résultats en classification est le critère utilisé ici pour la sélection d'un nombre suffisant de points au vu de la dimensionalité : dès lors que les résultats de

classification obtenus ont un écart type inférieur à 1 %, le nombre d'images nécessaire à cette performance est retenu et conservé comme un nombre suffisant d'images.
En pratique, dans le cas des expériences menées, l'écart type des résultats diminue assez rapidement pour qu'un nombre raisonnable d'images (en termes de temps de calcul) soit suffisant.

3.3. Réduction de la dimensionalité : sélection de caractéristiques par Forward avec OP-ELM

Une fois qu'une estimation d'un nombre suffisant de points (images) pour les données, a été obtenue, la réduction de dimensionalité peut être effectuée. La seconde étape de la Figure 4, un algorithme Forward utilisé avec un classifieur OP-ELM, est appliquée.

3.3.1. Algorithme de sélection Forward

L'algorithme Forward est un algorithme glouton [19] sélectionnant les caractéristiques une à une, en retenant la caractéristique donnant les meilleurs résultats une fois combinée à celles déjà séléctionnées. L'algorithme est présenté ci-dessous, avec x^i la i-ème caractéristique :

Algorithme 1 Forward

$\mathbf{R} = \{x^i, i \in [\![1,d]\!]\}$
$\mathbf{S} = \varnothing$
while $\mathbf{R} \neq \varnothing$ **do**
  for $x^j \in \mathbf{R}$ **do**
    Evaluer les performances avec $\mathbf{S} \cup x^j$
  end for
  Faire $\mathbf{S} = \mathbf{S} \cup \{x^k\}, \mathbf{R} = \mathbf{R} - x^k$ avec x^k la caractéristique donnant les meilleurs résultats dans la boucle
end while

Cet algorithme requiert donc $\frac{d(d-1)}{2}$ itérations pour se terminer – à comparer aux $2^d - 1$ itérations pour une recherche exhaustive des meilleures variables –, ce qui peut amener aux limites des temps de calculs possibles, selon la vitesse d'exécution du classifieur utilisé et le nombre de caractéristiques (dimensionalité). La rapidité d'exécution de l'OP-ELM est ici un avantage certain.
Malgré cette rapidité et ses bonnes performances dans la sélection des caractéristiques, la technique du Forward possède certains inconvénients. D'abord, deux caractéristiques donnant de bons résultats ensemble, mais n'ayant que peu d'intérêt séparées, peuvent ne pas être séléctionnées par le Forward.
D'autre part, le Forward ne permet pas de « repartir en arrière » dans le processus de sélection ; si les performances diminuent avec l'ajout de nouvelles caractéristiques, le Forward ne « désélectionnera » pas des caractéristiques qui ont pu être mal choisies ou inutiles par rapport à la sélection actuelle.
Néanmoins les diverses expériences utilisant le Forward ont prouvé que les résultats restent très bons et les sélections de variables sont effectivement intéressantes et pertinentes.

3.4. Méthodologie générale

La méthodologie générale schématisée en Figure 4 utilise donc tout d'abord un Boostrap avec 100 itérations pour des résultats fiables et une bonne estimation de l'écart type de ces résultats. Une fois un nombre suffisant d'images (pour la dimensionalité du problème, donc le nombre de caractéristiques) trouvé par cette étape, la sélection de caractéristiques est effectuée par le biais de l'algorithme Forward, afin de réduire la dimensionalité et mettre en évidence les faiblesses et sensibilités de l'algorithme de stéganographie utilisé par rapport aux caractéristiques.
Cette approche diffère de celle présentée dans les articles originaux présentant une méthodologie plus ou moins similaire [14,15]. En effet, le but de cette méthodologie et de cette étude est plus orienté vers la précision des résultats, afin que les performances présentées soient fiables. Le second objectif, déjà évoqué, est la sélection de caractéristiques et le double bénéfice de la diminution de la taille des données ainsi que de la possible interprétation de ces caractéristiques sélectionnées. Bien que les étapes de sélection de caractéristiques soient en général assez longues en termes de temps de calculs, le choix de l'algorithme Forward et du classifieur OP-ELM, permet ici d'obtenir une méthodologie rapide.
Les expériences proposées sont détaillées dans la section suivante, ainsi qu'une présentation des résultats principaux.

4. Expériences et Résultats

4.1. Protocole

4.1.1. Algorithmes de stéganographie utilisés

Six algorithmes différents ont été utilisés pour les expériences, afin de valider les résultats sur un panel relativement large de méthodes de stéganographie. Les algorithmes ont été choisis en raison de leurs performances bien connues (plus ou moins bonnes) : F5 [24], Model-Based (MBSteg) [20], MMx [9], JP Hide and Seek [10], OutGuess [17] et StegHide [6]. Chacun de ces algorithmes a été utilisé avec quatre taux d'insertion différents : 5, 10, 15 et 20 %.

4.1.2. Création de la base d'images

La base d'images est constituée de 13 000 images de scènes naturelles provenant de 6 appareils photo différents. Les images

problématiques (c'est-à-dire qui provoquent des valeurs très inhabituelles de caractéristiques) ont été enlevées de l'ensemble original de 16 000 images (photos complètement floues, images très unies de ciel ou d'océan...).
Les images ont ensuite été redimensionées en 800×600 (multiples de 8) afin d'éviter les possibles effets de bloc dus à la recompression JPEG sur une grille différente. Elles sont également transposées vers un espace de couleur en niveaux de gris, et finalement enregistrées sous un format sans pertes (pgm pour notre étude). Ce format sans perte est utilisé pour éviter toute recompression de l'image lors de l'opération de découpage. Les images sont découpées au format 512×512 et finalement enregistrées en JPEG, avec un facteur de qualité de 80 % (notre procédé d'extraction utilise des images de cette taille et avec ce facteur de qualité), dans l'espace de couleurs YCbCr.
Même si les expériences suivantes, ainsi que la méthodologie générale sont effectuées avec une telle taille d'image, n'importe quelle taille conviendrait.

4.1.3. Extraction des caractéristiques

Au final, l'ensemble des images est séparé en deux sous-ensembles égaux ; l'un d'eux reste intouché, tandis que l'autre est stéganographié avec les six algorithmes de stéganographie, pour les quatres taux d'insertions possibles : 5 %, 10 %, 15 % et 20 %.

4.2. Résultats et discussion

Les résultats sont présentés en suivant les étapes de la méthodologie.

4.2.1. Détermination d'un nombre suffisant de points

Comme présenté dans la première partie de la méthodologie, l'étape de Bootstrap avec 100 répétitions est utilisée avec des sous-ensembles de 100 à 3800 images. La Figure 6 illustre les résultats sur l'évolution de la valeur de l'écart type en fonction du nombre d'images utilisé.
Les tracés ne vont pas au delà de 3800 points pour deux raisons : les temps de calculs pour des ensembles plus importants commencent à devenir très importants, et d'autre part, l'écart type des résultats est déjà relativement faible et ne va aller qu'en diminuant. Il apparait que l'écart type dépend du nombre d'images utilisé, et l'on voit que lorsque l'on utilise 3800 images, il est toujours inférieur à 1 % de la meilleure valeur de classification, pour tous les cas considérés. Les résultats peuvent donc être considérés fiables avec une telle taille d'ensemble d'images.
La Figure 6 permet de voir que l'écart type des résultats diminue avec l'augmentation du nombre d'images, pour les algorithmes JPHS, MBSteg, OutGuess, StegHide et F5 : la valeur décroit et reste au final entre 0 et 0.5 % de la meilleure performance de classification, avec un nombre d'images de 3800.
Avec un nombre très faible d'images (100, par exemple), l'écart type des résultats se situe entre 1 et 5 % de la meilleure performance. Ceci signifie que des résultats présentés avec un si faible nombre d'images sont dans un intervalle de confiance de $\pm 5\%$.
Bien que la valeur de l'écart type diminue rapidement avec le nombre d'images, elle reste assez importante en dessous de 2000 images. On peut également remarquer que le taux d'insertion joue un rôle sur l'écart type, puisque ce dernier augmente lorsque le taux d'insertion diminue.
En effet, alors que les taux d'insertion de 15 et 20 % ne présentent pas de grandes différences dans les résultats, pour les quatres algorithmes mentionnés précédemment, il y a un écart important entre les taux 5-10 % et 20 %. Ceci est une conséquence des performances des algorithmes de stéganographie : un faible taux d'insertion est plus difficile à détecter, ce qui mène à une plage plus importante de performances que lorsque le taux d'insertion est important. La Figure 7 illustre ce comportement pour les cas de JPHS, StegHide, OutGuess et F5.
Enfin, l'algorithmes MM3 a un comportement « erratique », sur la Figure 6. Ce comportement peut s'expliquer par les très bons résultats du classifieur, proches de 100 % de bonne classification, ce qui rend l'écart type des résultats très faible : entre 0.07 % et 0 pour MM3. Les tracés sont donc erratiques en raison de ces valeurs très faibles.
Pour le reste de la méthodologie, l'étape de sélection de variables, le nombre suffisant d'images retenu est de 3800, le maximum.

4.2.2. Sélection de caractéristiques par Forward

Selon la méthode du Forward, les caractéristiques ont été classées par ordre d'importance au vu du problème de classification pour la stéganalyse. La Figure 7 trace le pourcentage de bonne classification en fonction du nombre de caractéristiques (classées par le Forward, toujours).
À nouveau, le cas de MM3 est à séparer des autres : quelque soit le taux d'insertion utilisé, la stéganalyse donne de très bons résultats. Avec seulement 4 caractéristiques pour ces deux cas, le pourcentage de bonne classification atteint les 100 %, même pour un taux d'insertion aussi faible que 5 %.
Les cinq autres algorithmes donnent des résultats différents :
- JPHS atteint rapidement un plateau (à 16 caractéristiques) pour tous les taux d'insertion ;
- OutGuess a également un plateau, mais à 25 caractéristiques, cette fois ;
- Le cas de F5 est similaire, avec un plateau à 20 caractéristiques ;
- Les performances pour l'algorithme StegHide peuvent être considérés comme maximales (dans l'intervalle de confiance), pour 40 caractéristiques, même si pour des taux d'insertion de 5 et 10%, 25 caractéristiques sont suffisantes pour atteindre ce résultat.

Enfin, l'algorithme MBSteg se comporte différemment des autres : les performances de classification sont en effet à leur maximum dès 25 caractéristiques à 5 % ; les autres taux d'insertion rendent nécessaire environ 70 caractéristiques pour arriver à des performances stables. La difficulté de la tâche de stéganalyse

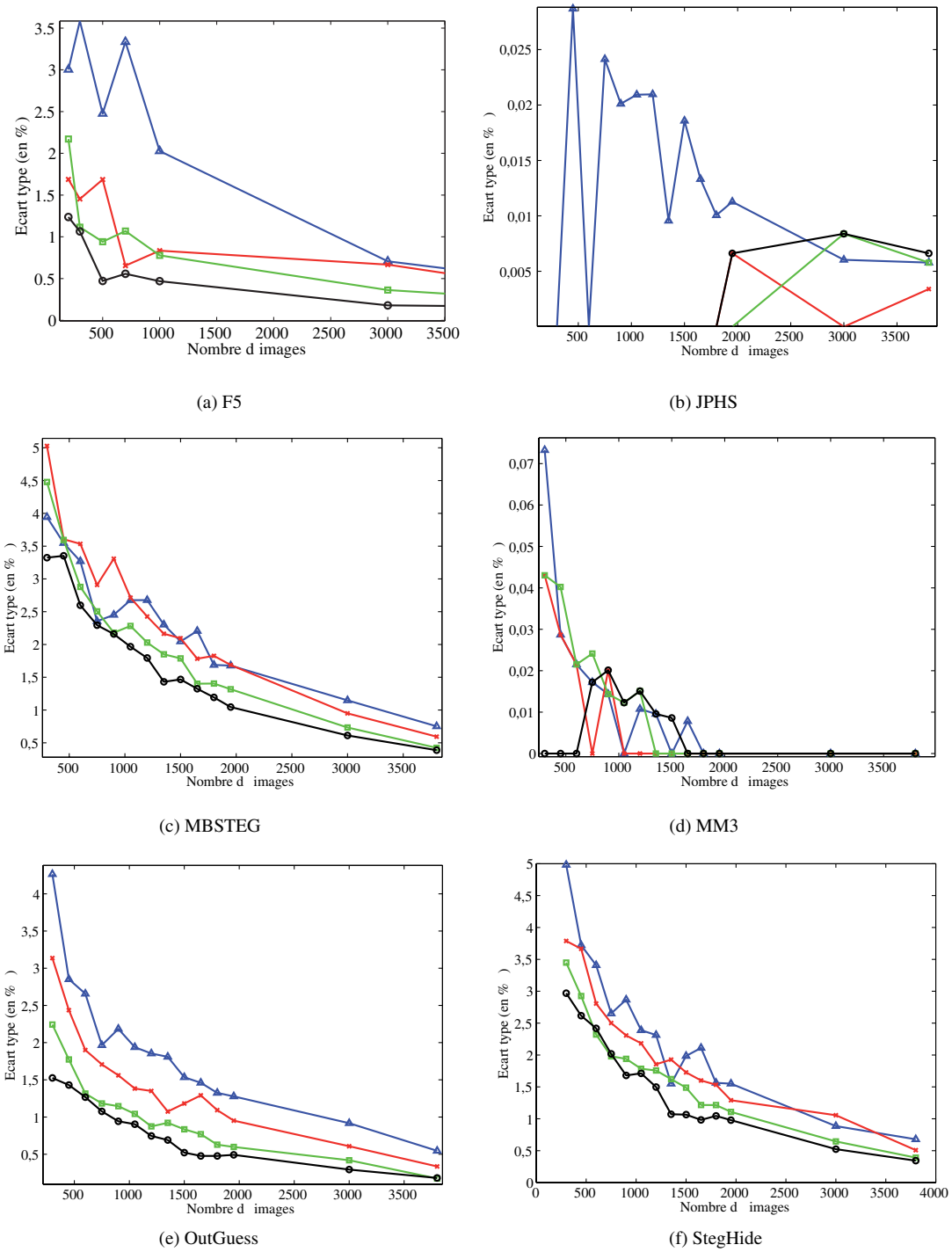
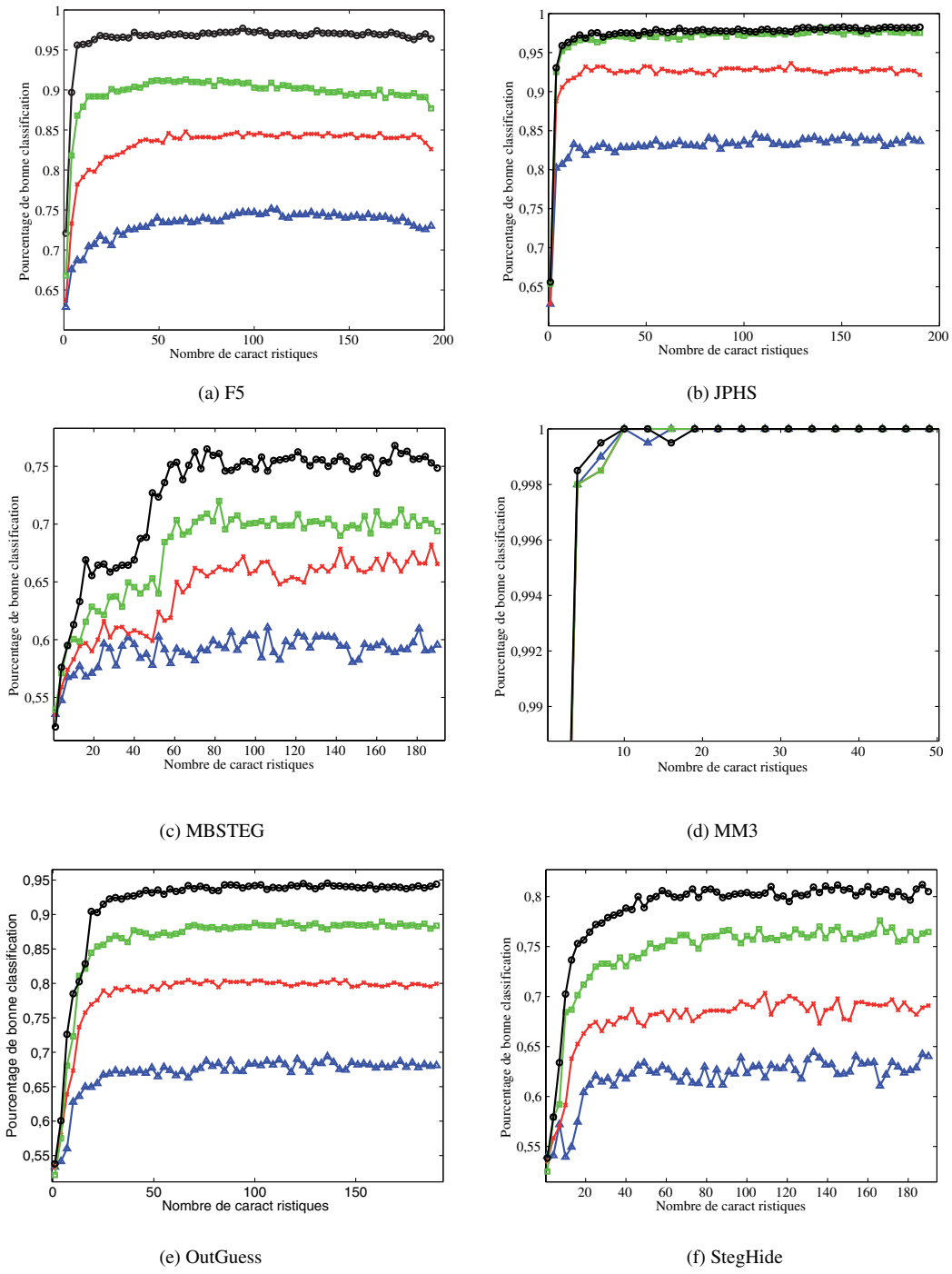


Figure 6. Ecart type en pourcentage de la meilleure valeur de classification en fonction du nombre d'images utilisé, pour les six algorithmes de stéganographie, aux quatres taux d'insertion : cercles noirs (◯) pour 20 %, carrés verts (□) pour 15 %, croix rouges (×) pour 10 % et triangles bleus (△) pour 5 %.

en général pour cet algorithme peut expliquer cette situation : les performances ne dépassent pas 77 % pour le cas à 20 % de taux d'insertion, ce qui reste 5 % derrière le cas de StegHide. Il est également possible que les limites mentionnées de l'algorithme Forward se retrouvent ici : les performances stagnent relativement entre 20 et 50 caractéristiques, et augmentent brutalement lorsque deux caractéristiques sont introduites, au delà de 50. Ces deux caractéristiques causant ce saut de performances sont les mêmes pour les trois taux d'insertion : la huitième composante de l'histogramme global $\mathbf{H}$ et la cinquième composante de l'histogramme dual $\mathbf{g}^{-2}$ (les détails sur ces notations se situent dans la section 4.4). Il est donc possible que ces deux caractéristiques se combinent avec d'autres sélectionnées précédemment pour créer cette augmentation de performances.



(a) F5 (b) JPHS

(c) MBSTEG (d) MM3

(e) OutGuess (f) StegHide

Figure 7. Pourcentages de bonne classification pour les six algorithmes de stéganographie en fonction du nombre de caractéristiques (pour les quatre taux d'insertion utilisés) : cercles noirs (◯) pour 20 %, carrés verts (□) pour 15 %, croix rouges (×) pour 10 % et triangles bleus (△) pour 5 %.

4.3. Discussion

Une analyse quantitative des caractéristiques séléctionnées est proposée ici : de par la méthode d'extraction des 193 caractéristiques (et leur nature), chacune peut être interprétée comme une propriété intrinsèque de l'image/de l'effet de l'algorithme de stéganographie sur la structure JPEG de l'image. Il n'est pas question de chercher de façon exhaustive toutes les faiblesses et de déduire de cette sélection le fonctionnement précis de l'algorithme de stéganographie, mais plutôt de comprendre certains phénomènes apparaissant en raison des procédés utilisés par cet algorithme. D'autre part, en raison de la redondance de l'information au sein de différentes caractéristiques (plusieurs caractéristiques peuvent détecter le même phénomène dû à la stéganographie), la sélection d'un nombre réduit de caractéristiques et son effet sur les performances finales est analysé ici.

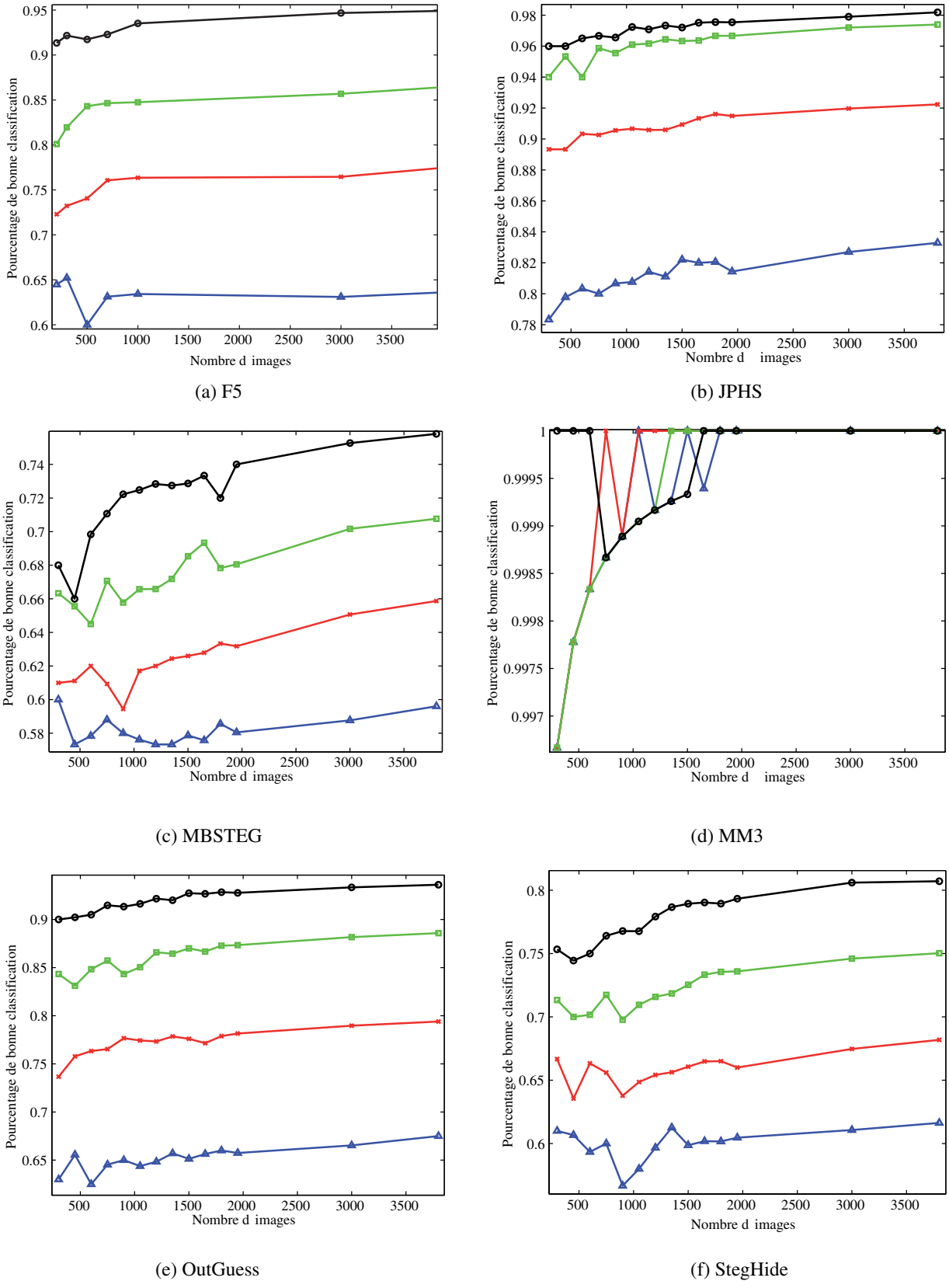


Figure 8. Pourcentages de bonne classification pour les six algorithmes de stéganographie en fonction du nombre du nombre d'images (pour les quatre taux d'insertion utilisés) : cercles noirs (◯) pour 20 %, carrés verts (□) pour 15 %, croix rouges (×) pour 10 % et triangles bleus (△) pour 5 %.

4.3.1. Performances avec des ensembles réduits

Les ensembles évoqués précédemment, propres à chaque algorithme et taux d'insertion, sont comparés en termes de performances de bonne classification, en Table 1.

Le but premier de la sélection de variables n'étant pas de privilégier des performances maximales, il est normal que les pourcentages obtenus avec les ensembles réduits soient potentiellement inférieurs aux valeurs maximales. Toutefois, au sein de l'intervalle de confiance pour 3800 images (écart type de 1 %), les performances restent sensiblement les mêmes.

Même si des performances supérieures à celles présentées pour l'ensemble réduit choisi sont atteignables, ce n'est pas le but de la sélection de caractéristiques proposée ici : seule la conservation de performances proches et équivalentes à la meilleure valeur, obtenue avec un nombre le plus réduit possible de caractéristiques, compte.

Il apparait donc que les ensembles réduits donnent globalement les mêmes résultats que l'ensemble complet des 193. Les cas de OutGuess et StegHide restent légèrement différents (résultats moins bons pour les ensembles réduits que pour l'ensemble complet), même s'il faut noter que ce sont les deux algorithmes ayant le plus grand écart type dans les résultats... Au final, les résultats peuvent donc être considérés comme identiques, dans l'intervalle de confiance, pour ensemble réduit et ensemble complet.

4.3.2. Intersections d'ensembles

Les ensembles réduits obtenus pour les différents taux d'insertions comportent certaines similarités. L'idée est ici d'obtenir un ensemble commun, le plus petit possible, tout en conservant les mêmes performances pour tous les taux d'insertion. L'intersection simple des ensembles réduits a été choisie pour construire ces nouveaux ensembles communs aux différents taux d'insertion (pour un même algorithme). L'intersection a été choisie – par rapport à une union, par exemple – afin de conserver le but de la sélection de caractéristiques : réduire autant que possible l'ensemble des caractéristiques utilisé, tout en conservant de bonnes performances.

Les ensembles réduits présentés dans le paragraphe précédent ont été utilisés pour constituer les intersections. Ainsi, les 14 premières caractéristiques obtenues pour F5 (pour chaque taux d'insertion), les 16 premières pour JPHS, et ainsi de suite, comme mentionné en Table 1.

Les résultats pour chaque algorithme et taux d'insertion sont présentés en Table 2, avec la taille de l'ensemble intersection.

On peut remarquer trois situations :

- Pour les cas de MM3, comme déjà observé, la stéganalyse est très aisée, même pour un taux d'insertion aussi faible que 5 %. Les résultats utilisant les ensembles réduits ou intersection sont toujours aussi bons que pour l'ensemble complet.
- MBSteg présente une situation assez différente des autres. Un nombre relativement faible de caractéristiques est suffisant pour de faibles taux d'insertion, tandis que les taux d'insertion plus importants nécessitaient environ 70 caractéristiques. L'ensemble intersection ne comporte que 23 caractéristiques, et ne contient pas les deux caractéristiques qui provoquent l'augmentation importante de performances. Ceci en raison de leur classement « éloigné » par le Forward.
- Les cas de JPHS, OutGuess, StegHide et F5 possèdent un comportement similaire, même si l'intersection obtenue pour StegHide est moins performante : les performances restent infé-

Table 1. Performances de l'OP-ELM (Leave-One-Out) pour l'ensemble complet de caractéristiques (193) et les performances en utilisant l'ensemble réduit (entre parenthèses) ; la taille de l'ensemble réduit est précisé (#).

	5 %	#	10 %	#	15 %	#	20 %	#
F5	71,82 (73,31)	46	82,36 (83,93)	38	88,35 (90,55)	33	96,12 (96,31)	15
JPHS	83.65 (83.25)	16	92.15 (93.20)	16	97.54 (96.60)	16	98.25 (97.50)	16
MBSteg	59.55 (59.70)	25	66.55 (66.20)	70	69.40 (70.20)	70	74.85 (76.25)	70
MM3	100.0 (99.95)	14	100.0 (100.0)	14	100.0 (100.0)	14	100.0 (100.0)	14
OutGuess	68.09 (67.30)	25	79.95 (78.95)	25	88.36 (86.90)	25	94.40 (92.45)	25
Steghide	64.03 (62.10)	25	69.10 (67.45)	25	76.45 (74.00)	40	80.50 (78.85)	40

Table 2. Performances de l'OP-ELM (Leave-One-Out) pour l'ensemble des 193 caractéristiques et l'ensemble intersection (entre parenthèses) ; la taille de l'ensemble intersection est également précisée.

	5 %	10 %	15 %	20 %	#
F5	71,82 (70,27)	82,36 (80,33)	88,35 (88,55)	96,12 (95,91)	12
JPHS	83.65 (82.25)	92.15 (90.57)	97.54 (95.33)	98.25 (95.62)	11
MBSteg	59.55 (58.70)	66.55 (60.70)	69.40 (62.51)	74.85 (64.60)	23
MM3	100.0 (99.98)	100.0 (99.95)	100.0 (100.0)	100.0 (100.0)	12
OutGuess	68.09 (67.33)	79.95 (78.35)	88.36 (86.04)	94.40 (92.50)	21
Steghide	64.03 (60.40)	69.10 (67.05)	76.45 (72.09)	80.53 (75.65)	15

rieures à celles obtenues avec l'ensemble complet, et cette situation tend à empirer lorsque le taux d'insertion augmente.
Au vu des ensembles pour les différents taux d'insertion, ceux obtenus pour un taux de 5 % sont très différents des autres. Les tracés de la Figure 7 permettent déjà de se douter de ce fait : l'évolution pour les taux 10,15 et 20 % sont relativement identiques, comparés avec le cas des 5 %, au sein d'un même algorithme.
Il apparaît donc que ce type d'ensembles communs pour un algorithme donné ne doive être construit qu'à partir d'ensembles obtenus pour des taux d'insertion significatifs. Les faibles taux d'insertion tendent en effet à rendre le procédé de sélection de caractéristiques moins fiable et les interprétations difficiles sinon impossibles.

4.4. Analyse des intersections d'ensembles de caractéristiques

Pour les raisons décrites ci-dessus, seules les intersections obtenues avec les ensembles pour les taux d'insertion 10, 15 et 20 % sont utilisées pour cette analyse. La sélection peut alors être considérée comme plus fiable et met mieux en valeur les possibles faiblesses des algorithmes étudiés. L'analyse suivante est une interprétation possible des ensembles de caractéristiques obtenus. Les conclusions et détails obtenus concernant les algorithmes de stéganographie au travers des caractéristiques ne sont en aucun cas absolus ni exhaustifs, mais ont pour but d'identifier une partie des faiblesses dans le but de pouvoir consolider et améliorer l'algorithme.
On peut toutefois remarquer qu'en raison de la méthode utilisée pour la sélection de caractéristiques (algorithme Forward), cette séléction n'est clairement pas *la* meilleure – qui permettrait de tirer des conclusions exactes et exhaustives concernant la méthode de stéganographie considérée –, mais une bonne sélection néanmoins permettant une analyse pertinente.
Tout d'abord, les notations de l'ensemble de caractéristiques utilisé [16] sont données, pour l'ensemble initial de 23 caractéristiques, dans la Table 3 :

Table 3. Les 23 caractéristiques.

Fonctionelle/ Caractéristique	Fonctionelle **F**
Histogramme Global	$\mathbf{H}/\|\|\mathbf{H}\|\|$
Histogramme Individuel pour 5 Modes DCT	$\mathbf{h}^{21}/\|\|\mathbf{h}^{21}\|\|$, $\mathbf{h}^{12}/\|\|\mathbf{h}^{12}\|\|$, $\mathbf{h}^{13}/\|\|\mathbf{h}^{13}\|\|$, $\mathbf{h}^{22}/\|\|\mathbf{h}^{22}\|\|$, $\mathbf{h}^{31}/\|\|\mathbf{h}^{31}\|\|$
Histogramme dual pour 11 valeurs DCT	$\mathbf{g}^{-5}/\|\|\mathbf{g}^{-5}\|\|$, $\mathbf{g}^{-4}/\|\|\mathbf{g}^{-4}\|\|$, ..., $\mathbf{g}^{4}/\|\|\mathbf{g}^{4}\|\|$, $\mathbf{g}^{5}/\|\|\mathbf{g}^{5}\|\|$
Variation	$\mathbf{V}$
Facteurs de bloc $L1$ et $L2$	$\mathbf{B}_1$, $\mathbf{B}_2$
Co-occurrence	N_{00}, N_{01}, N_{11}

Cet ensemble de 23 caractéristiques a été étendu à l'ensemble utilisé de 193, en supprimant la norme L_1 et en conservant l'ensemble des valeurs des matrices et vecteurs, comme décrit dans [16]. Les notations définitives des caractéristiques sont les suivantes :
- Histogramme global de 11 dimensions $\mathbf{H}(i), i = [\![1,11]\!]$
- 5 Histogrammes de coefficients DCT basse fréquence (11 dimensions chacun) $\mathbf{h}^{21}(i) \dots \mathbf{h}^{31}(i), i = [\![1,11]\!]$
- 11 histogrammes duaux (9 dimensions chacun)
$$\mathbf{g}^{-5}(i) \dots \mathbf{g}^{5}(i), i = [\![1,9]\!]$$
- Variation (1 dimension) $\mathbf{V}$
- 2 facteurs de bloc de dimension 1 $\mathbf{B}_1, \mathbf{B}_2$
- Matrice de co-occurrence de dimension 5×5
$$\mathbf{C}_{i,j}, i = [\![-2,2]\!], j = [\![-2,2]\!]$$

Les tables d'intersections d'ensembles réduits de caractéristiques sont présentées ci-après, avec une analyse pour chaque algorithme.

4.4.1. F5 et MM3

Table 4. Ensemble commun de caractéristiques (12) pour F5 (Intersection des 14 premiers pour 10,15 et 20 % de taux d'insertion).

$\mathbf{h}^{21}(3)$	$\mathbf{h}^{21}(6)$	$\mathbf{h}^{21}(7)$	$\mathbf{h}^{12}(3)$	$\mathbf{h}^{12}(5)$	$\mathbf{h}^{12}(7)$
$\mathbf{h}^{12}(9)$	$\mathbf{h}^{22}(6)$	$\mathbf{h}^{22}(9)$	$\mathbf{C}_{-2,-2}$	$\mathbf{C}_{-1,+1}$	$\mathbf{C}_{+0,+2}$

Table 5. Ensemble commun de caractéristiques (12) pour MM3 (Intersection des 14 premiers pour 10,15 et 20 % de taux d'insertion).

$\mathbf{h}^{21}(6)$	$\mathbf{h}^{21}(7)$	$\mathbf{h}^{12}(5)$	$\mathbf{h}^{12}(7)$	$\mathbf{h}^{22}(6)$	$\mathbf{g}^{-5}(1)$
$\mathbf{C}_{-2,-1}$	$\mathbf{C}_{-1,+1}$	$\mathbf{C}_{-1,+2}$	$\mathbf{C}_{+0,-2}$	$\mathbf{C}_{+0,+2}$	$\mathbf{C}_{+1,+1}$

Les cas F5 et MM3 sont à nouveau proches, probablement en raison de leur fonctionnement très similaire (basé, notamment, sur l'encodage matriciel), et possèdent une liste de caractéristiques sélectionnés (Tables 5, 4) assez proche. Leur faible résistance à la stéganalyse de MM3 vient principalement du fait que les histogrammes de coefficients DCT ne soient pas préservés. Les coefficients extrêmes $(-3,+3)$ sont sélectionnés pour F5, tandis que des plus faibles $(-1,+1)$ le sont pour MM3.
Ce qui rend la stéganalyse aisée
- Les coefficients DCT extrêmes pris en compte pour F5, et les plus faibles (en valeurs absolues), pour MM3.
- Un nombre faible de caractéristiques sélectionné (14 dans chaque cas).

4.4.2. JPHS

Pour le cas de JPHS (Table 6), un nombre faible de caractéristiques a également été sélectionné (16). JPHS ne préserve pas les coefficients basses et moyennes fréquences, ainsi que la cohérence fréquentielle (de la matrice de co-occurrence).

Tableau 6. Ensemble commun de caractéristiques (11) pour JPHS (Intersection des 16 premiers pour 10,15 et 20 % de taux d'insertion).

$\mathbf{h}^{12}(7)$	$\mathbf{h}^{12}(8)$	$\mathbf{h}^{13}(6)$	$\mathbf{h}^{22}(6)$	$\mathbf{h}^{31}(6)$	$\mathbf{h}^{12}(7)$
$\mathbf{C}_{-2,-1}$	$\mathbf{C}_{-1,+1}$	$\mathbf{C}_{-1,+2}$	$\mathbf{C}_{+0,+1}$	$\mathbf{C}_{+0,+2}$	

Ce qui rend la stéganalyse aisée
- L'histogramme des coefficients DCT est conservé.
- La cohérence fréquentielle n'est pas conservée, même pour des valeurs proches.
- Le nombre de caractéristiques retenues est faible.

4.4.3. MBSteg

Table 7. Ensemble commun de caractéristiques (46) pour MBSteg (Intersection des 70 premiers pour 10, 15 et 20 % de taux d'insertion).

$\mathbf{H}(4)$	$\mathbf{H}(6)$	$\mathbf{H}(8)$	$\mathbf{h}^{21}(5)$	$\mathbf{h}^{21}(6)$	$\mathbf{h}^{21}(8)$	$\mathbf{h}^{12}(4)$	$\mathbf{h}^{12}(5)$
$\mathbf{h}^{12}(6)$	$\mathbf{h}^{12}(1)$	$\mathbf{h}^{13}(3)$	$\mathbf{h}^{13}(4)$	$\mathbf{h}^{13}(5)$	$\mathbf{h}^{13}(6)$	$\mathbf{h}^{13}(7)$	$\mathbf{h}^{13}(8)$
$\mathbf{h}^{22}(3)$	$\mathbf{h}^{22}(4)$	$\mathbf{h}^{22}(6)$	$\mathbf{h}^{22}(9)$	$\mathbf{h}^{31}(3)$	$\mathbf{h}^{31}(4)$	$\mathbf{h}^{31}(6)$	$\mathbf{h}^{31}(8)$
$\mathbf{h}^{31}(10)$	$\mathbf{g}^{-5}(1)$	$\mathbf{g}^{-5}(4)$	$\mathbf{g}^{-5}(5)$	$\mathbf{g}^{-3}(1)$	$\mathbf{g}^{-1}(1)$	$\mathbf{g}^{-1}(2)$	$\mathbf{g}^{-1}(6)$
$\mathbf{g}^{2}(1)$	$\mathbf{g}^{4}(3)$	$\mathbf{C}_{-2,-2}$	$\mathbf{C}_{-2,-1}$	$\mathbf{C}_{-2,+0}$	$\mathbf{C}_{-2,+1}$	$\mathbf{C}_{-2,+2}$	$\mathbf{C}_{+0,+0}$
$\mathbf{C}_{+0,+2}$	$\mathbf{C}_{+1,-2}$	$\mathbf{C}_{+1,-1}$	$\mathbf{C}_{+2,+0}$	$\mathbf{C}_{+2,+2}$	$\mathbf{B}_1$		

MBSteg nécessite un nombre important de caractéristiques (Table 7) pour obtenir des performances acceptables. Ceci signifie que l'algorithme est particulièrement difficile à détecter (pour le cas de ces 193 caractéristiques de Fridrich). Les caractéristiques notables sont par exemple celles d'histogramme global pour les valeurs 0, –2 et 2, apparaissant en raison du processus de calibration des caractéristiques de Fridrich. MBSteg préserve les coefficients des histogrammes, mais ne prend pas en compte la calibration, ce qui cause ce comportement par rapport aux histogrammes.
Ce qui rend la stéganalyse aisée
- Le processus de calibration permet de mettre en évidence la stéganographie clairement.

4.4.4. OutGuess

Principalement des valeurs extrêmes d'histogrammes sont utilisées (–2, –1) pour OutGuess (Table 8). Le processus de calibration a également été utile dans ce cas, puisque l'histogramme de la valeur 0 a été pris en compte. Les valeurs de co-occurrence entre –2 et –1 sont également d'importance.

Table 8. Ensemble commun de caractéristiques (20) pour OutGuess (Intersection des 25 premiers pour 10, 15 et 20 % de taux d'insertion).

$\mathbf{h}^{21}(4)$	$\mathbf{h}^{21}(5)$	$\mathbf{h}^{21}(7)$	$\mathbf{h}^{12}(4)$	$\mathbf{h}^{12}(5)$	$\mathbf{h}^{13}(4)$	$\mathbf{h}^{13}(5)$	$\mathbf{h}^{13}(6)$	$\mathbf{h}^{22}(4)$	$\mathbf{h}^{22}(6)$
$\mathbf{h}^{31}(4)$	$\mathbf{h}^{31}(5)$	$\mathbf{h}^{31}(6)$	$\mathbf{g}^{-2}(1)$	$\mathbf{g}^{-2}(2)$	$\mathbf{C}_{-2,-1}$	$\mathbf{C}_{-2,+1}$	$\mathbf{C}_{-1,+0}$	$\mathbf{C}_{-1,+1}$	$\mathbf{C}_{+0,+2}$

Ce qui rend la stéganalyse aisée
- Les coefficients –2 et –1 sont clairement les points faibles d'OutGuess.

Néanmoins, un nombre relativement important de caractéristiques est utilisé (25) pour obtenir un résultat suffisant en classification, ce qui tend à montrer qu'OutGuess reste un algorithme relativement fiable (difficile à détecter par ce processus de stéganalyse avec un nombre faible de caractéristiques).

4.4.5. StegHide

Table 9. Ensemble commun de caractéristiques (21) pour StegHide (Intersection des 25 premiers pour 10 % et 40 premiers pour 15 et 20 % de taux d'insertion).

$\mathbf{h}^{21}(4)$	$\mathbf{h}^{21}(5)$	$\mathbf{h}^{12}(6)$	$\mathbf{h}^{12}(8)$	$\mathbf{h}^{13}(4)$	$\mathbf{h}^{13}(5)$	$\mathbf{h}^{13}(6)$
$\mathbf{h}^{22}(6)$	$\mathbf{h}^{31}(4)$	$\mathbf{h}^{31}(5)$	$\mathbf{h}^{31}(6)$	$\mathbf{h}^{31}(7)$	$\mathbf{h}^{31}(8)$	$\mathbf{g}^{-2}(1)$
$\mathbf{C}_{-2,-1}$	$\mathbf{C}_{-2,+1}$	$\mathbf{C}_{-1,+0}$	$\mathbf{C}_{-1,+1}$	$\mathbf{C}_{+0,+2}$	$\mathbf{C}_{+1,-2}$	$\mathbf{C}_{+1,-1}$

Pour StegHide, les histogrammes sont principalement utilisés (Table 9), avec des coefficients haute fréquence (31, 13, 22) et pour des valeurs faibles (entre –1 et +1). La matrice de co-occurrence est utilisée pour de faibles valeurs avec de grands écarts ($0 \leftrightarrow 2$, $-2 \leftrightarrow +1$, $+1 \leftrightarrow -2$).
Ce qui rend la stéganalyse aisée
- L'utilisation de hautes fréquences avec de faibles valeurs.
- Les valeurs de co-occurrence pour de faibles valeurs avec grands écarts.

De nouveau, StegHide apparait comme un algorithme difficile à stéganalyser, au vu du nombre de caractéristiques nécessaires à l'obtention de bonnes performances. De plus, les caractéristiques sélectionnées sont relativement inhabituelles, comparé aux autres algorithmes, et rendent une analyse plus difficile.

5. Conclusions et perspectives

Cet article propose une méthodologie permettant d'estimer l'écart type des résultats typiquement obtenus en stéganalyse, ainsi qu'une estimation d'un nombre suffisant d'images à utiliser afin d'obtenir des résultats fiables. Bien que cette méthodologie ait été proposée pour un ensemble de caractéristiques précis, elle peut être étendue à d'autres. La deuxième étape de la méthodologie tente de réduire le nombre de caractéristiques nécessaires – et d'en tirer une interprétation grâce à leur nombre réduit –, par une sélection des caractéristiques les plus efficaces et utiles pour le problème de classification. Un algorithme Forward permet cette sélection, et rend possible l'analyse des

caractéristiques les plus sensibles pour un algorithme donné, et donc une mise en évidence des possibles faiblesses et du fonctionnement de l'algorithme de stéganographie.

Cinq points majeurs ont été tirés de cette étude :

- Les résultats concernant les valeurs d'écart type pour les résultats de classification typiques ont montré que ce type de stéganalyse ne peut être considérée comme fiable que si un nombre suffisant d'images est utilisé pour l'entrainement du classifieur. Il apparait en effet que des résultats affichant une augmentation de quelques pourcents et donnés avec un écart type de 2 % sont peu « crédibles ».
- Les performances dépendent de façon importante du nombre d'images utilisé, comme présenté sur la Figure 8 (du moins pour les algorithmes ayant un comportement « normal »). Une autre constatation est que les comparaisons entre algorithmes, en termes de performances, doivent être faites à nombre d'images égal (pour l'entrainement du classifieur), au risque d'avoir un écart type très différent pour les deux résultats.
- Grâce à la seconde étape de la méthodologie, le nombre de caractéristiques nécessaire à une bonne classification, peut être diminué. Cette étape possède trois avantages : (1) les performances restent identiques si l'ensemble réduit a été correctement construit ; (2) les caractéristiques sélectionnées sont pertinentes pour le problème, et rendent une analyse possible ; (3) les faiblesses de l'algorithme de stéganographie considéré apparaissent plus clairement, de par cette sélection, et peuvent mener à des améliorations éventuelles de l'algorithme, en vu d'une plus grande sécurité.
- L'analyse des ensembles intersection tend à montrer que les algorithmes étudiés sont sensibles à des caractéristiques relativement similaires. Néanmoins, lorsque le taux d'insertion est de seulement 5 %, ou pour les algorithmes les plus sûrs, certaines caractéristiques particulières apparaissent.
- Le nombre de caractéristiques sélectionné conditionne les performances de l'algorithme, ce qui signifie qu'un nombre de caractéristiques sélectionné important tend à indiquer que l'algorithme est meilleur, en termes de sécurité. Une possible extension de cette idée est de vouloir modifier autant de caractéristiques indépendantes que possible, afin de forcer le stéganalyseur à utiliser un nombre important de caractéristiques et rendre la stéganalyse plus difficile. L'idée étant que les modifications sont alors réparties sur autant de caractéristiques différentes et indépendantes que possible.

Références

[1] R. BELLMAN, *Adaptive control processes: a guided tour*. Princeton University Press, 1961.

[2] C. CACHIN. An information-theoretic model for steganography. In *Information Hiding: 2nd International Workshop*, volume 1525 of *Lecture Notes in Computer Science*, pages 306-318, 14-17 April 1998.

[3] JPEG Comittee, http://www. jpeg. com.

[4] D. FRANÇOIS, *High-dimensional data analysis: optimal metrics and feature selection*. PhD thesis, Université catholique de Louvain, September 2006.

[5] J. FRIDRICH, Feature-based steganalysis for jpeg images and its implications for future design of steganographic schemes. In *Information Hiding : 6th International Workshop*, volume 3200 of *Lecture Notes in Computer Science*, pages 67-81, May 23-25 2004.

[6] S. HETZL and P. MUTZEL, A graph-theoretic approach to steganography. In Dittmann J., Katzenbeisser S., and Uhl A., editors, *CMS 2005*, Lecture Notes in Computer Science 3677, pages 119-128. Springer-Verlag, 2005.

[7] G.-B. HUANG, Q.-Y. ZHU, and C.-K. SIEW, Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1-3):489-501, December 2006.

[8] B. EFRON R.J. and TIBSHIRANI, *An Introduction to the Bootstrap*. Chapman et al., Londres, 1994.

[9] Y. KIM, Z. DURIC, and D. RICHARDS, Modified matrix encoding technique for minimal distortion steganography. In *Information Hiding 2007*, volume 4437/2007, pages 314-327, 2007.

[10] A. LATHAM, Jphide&seek, August 1999. http://linux01. gwdg. de/ alatham/stego. html.

[11] S. LYU and H. FARID, Detecting hidden messages using higher-order statistics and support vector machines. In *5th International Workshop on Information Hiding*, Noordwijkerhout, The Netherlands, 2002.

[12] D. MCCULLAGH, Secret Messages Come in . Wavs. Online Newspaper: Wired News, February 2001. http://www. wired. com-/news/politics/ 0,1283,41861,00. html.

[13] Y. MICHE, P. BAS, C. JUTTEN, O. SIMULA, and A. LENDASSE, A methodology for building regression models using extreme learning machine: OP-ELM. In *ESANN 2008, European Symposium on Artificial Neural Networks, Bruges, Belgium*, April 23-25 2008. to be published.

[14] Y. MICHE, P. BAS, A. LENDASSE, C. JUTTEN, and O. SIMULA, Extracting relevant features of steganographic schemes by feature selection techniques. In *Wacha'07: Third Wavilla Challenge*, June 14 2007.

[15] Y. MICHE, B. ROUE, P. BAS, and A. LENDASSE, A feature selection methodology for steganalysis. In *MRCS06, International Workshop on Multimedia Content Representation, Classification and Security, Istanbul (Turkey)*, Lecture Notes in Computer Science. Springer-Verlag, September 11-13 2006.

[16] T. PEVNY and J. FRIDRICH, Merging markov and dct features for multi-class jpeg steganalysis. In *IS&T/SPIE 19th Annual Symposium Electronic Imaging Science and Technology*, volume 6505 of *Lecture Notes in Computer Science*, January 29th - February 1st 2007.

[17] N. PROVOS, Defending against statistical steganalysis. In *10th USENIX Security Symposium*, pages 323-335, 13-17 April 2001.

[18] N. PROVOS and P. HONEYMAN, Detecting steganographic content on the internet. In *Network and Distributed System Security Symposium*. The Internet Society, 2002.

[19] F. ROSSI, A. LENDASSE, D. FRANÇOIS, V. WERTZ, and M. VERLEYSEN, Mutual information for the selection of relevant variables in spectrometric nonlinear modelling. *Chemometrics and Intelligent Laboratory Systems*, 80:215-226, 2006.

[20] P. SALLEE, Model-based steganography. In *Digital Watermarking*, volume 2939/2004 of *Lecture Notes in Computer Science*, pages 154-167. Springer Berlin/Heidelberg, 2004.

[21] Y.Q. SHI, C. CHEN, and W. CHEN, A markov process based approach to effective attacking jpeg steganography. In *ICME'06 : Internation Conference on Multimedia and Expo*, Lecture Notes in Computer Science, 9-12 July 2006.

[22] A. SORJAMAA, Y. MICHE, and A. LENDASSE, Long-term prediction of time series using nne-based projection and op-elm. In *IJCNN2008: International Joint Conference on Neural Networks*, June 2008. to be published.

[23] M. VERLEYSEN and D. FRANÇOIS, The curse of dimensionality in data mining and time series prediction. In *IWANN'05 : 8th International Work-Conference on Artificial Neural Network, volume 3512 of Lecture Notes in Computer Science*, pages 758-770, June 8-10 2005.

[24] A. WESTFELD, F5-a steganographic algorithm. In *Information Hiding: 4th International Workshop*, volume 2137, pages 289-302, 25-27 Avril 2001.

[25] A. WESTFELD and A. PFITZMANN, Attacks on steganographic systems. In *IH '99: Proceedings of the Third International Workshop on Information Hiding*, pages 61-76, London, UK, 2000. Springer-Verlag.

Yoan **Miche**

Yoan Miche was born in 1983 in France. He received an Ingeneer's Degree from Institut National Polytechnique de Grenoble (INPG, France), and more specifically from TELECOM, INPG, on September 2006. He also graduated with a Master's Degree in Signal, Image, Speak and Telecom from ENSERG, INPG, at the same time. He is currently working in both Gipsa-Lab, INPG, France and ICS Lab, TKK, Finland, as a Ph.D student. His main research interests are steganography/steganalysis and machine learning for classification.

Patrick **Bas**

Dr Patrick Bas received the Electrical Engineering degree from the Institut National Polytechnique de Grenoble, France, in 1997 and the Ph.D. degree in Signal and Image processing from Institut National Polytechnique de Grenoble, France, in 2000. From 1997 to 2000, he was a member of the Laboratoire des Images et des Signaux de Grenoble (LIS), France where he worked on still image watermarking. During his post-doctoral activities, he was a Member of the Communications and Remote Sensing Laboratory of the Faculty of Engineering at the Université Catholique de Louvain, Belgium. His research interests include synchronisation and security evaluation in watermarking, and steganalysis.

Amaury **Lendasse**

Amaury Lendasse was born in 1972 in Belgium. He received the M.S. degree in mechanical engineering from the Université catholique de Louvain (Belgium) in 1996, M.S. in control in 1997 and Ph.D. in 2003 from the same University. In 2003, he has been a postdoctoral researcher in the Computational Neurodynamics Lab at the University of Memphis. Since 2004, he is a senior researcher in the Adaptive Informatics Research Centre in the Helsinki University of Technology in Finland. He is leading the Time Series Prediction Group. He is the author or the coauthor of 110 scientific papers in international journals, books or communications to conferences with reviewing committee. His research includes time series prediction, chemometrics, variable selection, noise variance estimation, determination of missing values in temporal databases, nonlinear approximation in financial problems, functional neural networks and classification.

Christian **Jutten**

Christian Jutten received the Ph.D. degree in 1981 and the Docteur És Sciences degree in 1987 from the Institut National Polytechnique of Grenoble (France). He taught as associate professor in École Nationale Supérieure d'Electronique et de Radioélectricité of Grenoble from 1982 to 1989. He was a visiting professor in Swiss Federal Polytechnic Institute in Lausanne in 1989, he became full professor in the Sciences and Techniques Institute, Université Joseph Fourier of Grenoble. For 20 years, his research interests are source separation and independent component analysis and learning in neural networks. He has been associate editor of IEEE Trans. on Circuits and Systems (1994,1995), and co-organizer with Dr. J.-F. Cardoso and Prof. Ph. Loubaton of the 1st International Conference on Blind Signal Separation and Independent Component Analysis (Aussois, France, January 1999). He is currently member of a technical committee of IEEE Circuits and Systems society on blind signal processing.

Olli **Simula**

Olli Simula received Doctor of Science (Tech.) degree in computer science and engineering from Helsinki University of Technology (TKK), Finland, in 1979. Dr. Simula is Professor of Computer Science and Engineering at the Department of Information and Computer Science at Helsinki University of Technology, TKK. He is also Dean of the Faculty of Information and Natural Sciences at TKK. During the academic year 1977-78 Dr. Simula was a research fellow at Delft University of Technology, Delft, The Netherlands.