# La qualité de l'information dans les réseaux sociaux en ligne: une approche non supervisée et rapide de détection de spam

**Mahdi Washha , Manel Mezghani , Florence Sèdes**

*Institut de Recherche en Informatique de Toulouse (IRIT), Université de Toulouse, CNRS, INPT, UPS, UT1, UT2J, 31062 TOULOUSE Cedex 9, France*
*mahdi.washha,manel.mezghanni,florence.sedes@irit.fr*

RÉSUMÉ. *Les réseaux sociaux en ligne fournissent des données utiles pour une vaste gamme d'applications. Cependant, les interfaces faciles à utiliser et les faibles limites de sécurité à la publication génèrent divers problèmes liés à la qualité de "l'information", i. e. des contenus générés par l'utilisateur via ces réseaux. L'existence d'utilisateurs mal intentionnés, nommés spammeurs sociaux, entre dans ce cadre. Les principales limitations des approches de détection de spam sont qu'elles reposent sur des approches d'apprentissage supervisé qui exigent des ensembles de données vérité terrain. Par ailleurs, les approches de détection basées sur les comptes des utilisateurs ne sont pas utilisables pour le traitement de grandes collections de publications, car elles nécessitent des mois pour traiter ces collections. Dans cet article, nous présentons donc une approche d'apprentissage non supervisé dédiée à la détection de comptes spam dans de grandes collections de thématiques "tendances". Notre approche a été testée sur Twitter et a prouvé son efficacité ainsi que sa rapidité par rapport aux approches d'apprentissage supervisé.*

ABSTRACT. *Online social networks provide data valuable for a tremendous range of applications. However, the easy-to-use interfaces and low limits of the publication generate various information quality problems, such the user-generated content in such networks. The existence of ill-intentioned users, so-called social spammers, belongs to this environments. The major limitations of the methods detecting spams are the use of supervised learning approaches that requiring ground truth data-sets. Moreover, the account-based detection methods are not practical for processing "crawled" large collections of social posts, requiring months to process such collections. Hence, in this paper, we introduce a design of an unsupervised learning approach dedicated for detecting spam accounts existing in large collections of "trending" topics. Our experimental evaluation on Twitter demonstrates the efficiency of our approach as well as its speed comparing to supervised approaches.*

MOTS-CLÉS : *Twitter, Réseau Social, Spam*

KEYWORDS: *Twitter, Social Network, Spam*

## 1. Introduction

Les réseaux sociaux en ligne (OSN) sont devenus un moyen de communication puissant dans lequel les utilisateurs ont la possibilité de partager des liens, de discuter et de s'inter-connecter. Les interfaces faciles à utiliser et les faibles limites de sécurité à la publication ne contribuent pas à maintenir un niveau constant de qualité de l'information (QI). Ces caractéristiques ont rendu les OSN vulnérables à diverses attaques par un certain type d'utilisateurs mal intentionnés, appelés spammeurs sociaux. Les spammeurs sociaux affichent un contenu illicite ou non pertinent par rapport à un contexte donné ou une thématique particulière. Plus généralement, les spammeurs sociaux ont à disposition un large éventail tde echniques pour publier des contenus spam, résumés dans (Benevenuto *et al.*, 2010): (i) diffusion de publicités pour générer des ventes et des profits illégaux; (ii) diffusion de matériel pornographique; (iii) publication de virus et de malwares; (iv) création de sites Web de phishing pour révéler des informations sensibles, ...

Les impacts négatifs du spam social ne pouvant être ignorés: traiter ce problème contribue à améliorer la qualité de l'information en détectant et en filtrant les contenus "spam" existants dans les collections de données téléchargées ou dans les OSN. La présence du spam dans les OSN est à l'origine de nuisances majeures telles que: (i) polluer les résultats de recherche; (ii) dégrader l'exactitude des statistiques obtenues à travers des outils d'extraction d'information; (iii) consommer des ressources de stockage; (iv) violer la vie privée des utilisateurs, ... Les mécanismes anti-spam s'avèrent insuffisants pour mettre fin au problème de spam, ce qui suscite de réelles inquiétudes quant à la qualité des collections de données "aspirées". La solution proposée dans ce papier peut être intégrée à des recherches portant sur un large éventail de problèmes de OSN afin d'améliorer leurs résultats, comme par exemple les travaux de (Abascal-Mena *et al.*, 2015 ; Canut *et al.*, 2015) sur le profilage social et la détection des communautés socio-sémantiques, dans lesquesl des collections de données à grande échelle des thématiques tendances de Twitter servent de base aux expérimentations.
Dans la bataille de la détection du spam social sur Twitter, plusieurs approches ont été proposées pour détecter les campagnes de spam et les comptes spam individuels, mais peu d'efforts ont été dédiés aux tweets spam individuels. Ces approches fonctionnent en utilisant le concept d'extraction de caractéristiques combiné avec une approche d'apprentissage supervisé pour construire un modèle prédictif basé sur un jeu de données annoté (vérité terrain). Cependant, s'appuyer sur l'approche d'apprentissage supervisé dans la conception de modèles prédictifs devient moins efficace pour détecter les utilisateurs spammeurs ou du contenu spam en raison de l'évolution rapide du comportement de ces spammeurs sociaux. Ceux-ci adoptent des modèles de contenu dynamique dans les OSNs ainsi qu'un changement de leurs stratégies de spam pour se faire passer pour des utilisateurs "normaux". Les approches anti-spam statiques présentent donc un retard considérable en ne considérant pas l'évolution rapide des comportements des spammeurs sociaux.
Au-delà de cette dynamique évidente, les approches de détection actuelles basées sur les campagnes et sur les comptes individuels ne conviennent pas aux ensembles de données à grande échelle de thématiques de Twitter, nécessitant des mois pour traiter

ce volume de données. Comme les tweets se composent de méta-données simples (par exemple nom d'utilisateur, date de création), ces approches ont été conçues en fonction de caractéristiques avancées (par exemple, les tweets de l'utilisateur au fil du temps) nécessitant des informations supplémentaires des serveurs de Twitter. Ce dernier fournit des fonctions pour récupérer ces informations supplémentaires (par exemple, les abonnés de l'utilisateur - *followers* et les *followees*) sur un objet donné à l'aide des API REST [1]. Cependant, Twitter limite le nombre d'appels autorisés via l'API à une fenêtre de temps définie (par exemple, 40 appels en 15 minutes). La récupération d'informations, y compris les méta-données des *followers* et des *followees*, pour un demi-million d'utilisateurs qui ont posté un million de tweets, peut dès lors prendre environ trois mois.

Dans cet article, nous proposons une approche non supervisée pour filtrer les comptes spam dans des ensembles de données à grande échelle de thématiques tendances dans Twitter. Plus précisément, nous n'exploitons que les méta-données disponibles dans une thématique donnée, sans récupérer aucun type d'information des serveurs de Twitter.

Le reste de l'article est organisé comme suit. La section 2 présente le mécanisme anti-spam de Twitter ainsi que les approches de détection de spam social proposées dans la littérature. La section 3 introduit les notations, la formalisation des problèmes et la conception de notre approche. La section 4 détaille l'ensemble de données utilisées pour l'expérimenter et la valider. La configuration expérimentale et une série d'expérimentations évaluant l'approche proposée sont décrites dans la section 5. Enfin, la section 6 conclut le travail.

## 2. Contexte et Etat de l'art

Dans cette section, nous présentons le mécanisme utilisé dans Twitter pour combattre le spam ainsi que les approches de détection de spam.

**Mécanisme Anti-Spam de Twitter.** Twitter combat les spammeurs sociaux en permettant aux utilisateurs de signaler des comptes spam simplement en cliquant sur l'option "Signaler: ils publient du spam" disponible sur la page du compte. Lorsqu'un utilisateur signale un compte particulier, les administrateurs de Twitter examinent manuellement le compte rendu pour prendre la décision de suspension. Cependant, l'adoption d'une telle approche pour lutter contre les spammeurs nécessite des efforts considérables des utilisateurs et des administrateurs. Malheureusement, la probabilité de détecter et de suspendre un compte quelques jours après sa création est inférieure à 1%. Par conséquent, ces lacunes ont motivé les chercheurs à proposer des approches plus puissantes pour les applications qui utilisent Twitter comme source d'information à savoir les approches d'apprentissage automatique (*Machine learning approaches*) comme étant une approche entièrement automatisée que nous détaillons dans ce papier.

**Approche d'apprentissage automatique**. Ces approches sont construits en employant trois niveaux de détection:

1. https://dev.twitter.com/rest/public

*Niveau tweet:* À ce niveau, les tweets individuels sont vérifiés afin d'éliminer d'éventuels contenus indésirables. Benevenuto (Benevenuto *et al.*, 2010) a extrait un ensemble de caractéristiques statistiques simples du tweet telles que le nombre de mots, le nombre de hashtags et le nombre de caractères. Ensuite, un classifieur binaire est construit sur un petit ensemble de données annotées. Martinez-Romo et Araujo (Martinez-Romo, Araujo, 2013) ont détecté des tweets spam dans les thématiques tendances à travers l'utilisation de modèles de langage pour extraire plus de caractéristiques telles que la divergence de distribution de probabilité entre un tweet donné et d'autres tweets. Le problème majeur à ce niveau de détection provient du manque d'information qui peut être extraite du tweet lui-même. En outre, la construction de modèles de langues à l'aide de tweets dans les thématiques tendances échoue définitivement quand il y a d'énormes attaques de spam.

*Niveau compte:* Les approches conçues dans (Wang, 2010 ; Benevenuto *et al.*, 2010 ; Stringhini *et al.*, 2010 ; McCord, Chuah, 2011 ; Cao, Caverlee, 2015) construisent d'abord des vecteurs en extrayant des caractéristiques "à la main" telles que le nombre de *followers*, et l'intermédiation de noeuds. Ensuite, des algorithmes d'apprentissage automatique supervisés sont appliqués pour construire un modèle de classification sur un ensemble de données annotées. Malgré un taux de détection élevé en exploitant ces caractéristiques, les extraire est chronophage en raison du temps nécessaire au recueil des informations du serveur de Twitter via l'utilisation de l'API REST. En effet, ces API sont limitées à un certain nombre prédéfini d'appels, ce qui rend l'extraction de la plupart des caractéristiques impossible, en particulier dans le traitement de données à grande échelle.

*Niveau campagne:* Chu et al. (Chu, Widjaja, Wang, 2012) ont proposé une approche de détection de campagne de spam à travers le regroupement des comptes spam selon les URL disponibles dans les tweets. Un vecteur est ensuite représenté, via des caractéristiques similaires aux approches de détection au niveau du compte. Dans (Chu, Gianvecchio *et al.*, 2012), un modèle de classification a été conçu pour capturer les différences entre campagne, humain et *cyborg*. Malheureusement, ce niveau de détection présente des inconvénients similaires à ceux mentionnés pour le niveau "compte", ce qui rend ces solutions non évolutives pour de grandes collections d'utilisateurs ou de tweets.

Nous présentons dans la section suivante notre approche afin de surmonter les problématiques évoquées dans l'apprentissage automatique au niveau *tweet* .

## 3. Modèle Prédictif

Dans cette section, nous présentons les premières définitions et notations utilisées dans la modélisation de notre solution. Ensuite, nous présentons la conception de notre approche de détection des comptes spam.

### *3.1. Notations, définitions et formalisation du problème*

Le concept de *Topic* [2] ou thématique peut être défini comme une représentation de structures sémantiques cachées dans une collection de textes (par exemple: des documents textuels, des tweets). *Trending Topic* ou thématique tendance est un mot ou une expression (par exemple #TopChef) qui est mentionné à un taux plus élevé que d'autres. Les thématiques tendances sont automatiquement identifiées par un algorithme qui identifie des thématiques qui sont plus massivement diffusées par les utilisateurs que d'autres.

Comme plusieurs utilisateurs peuvent tweeter sur le même sujet, nous modélisons une thématique tendance particulière $T$ comme un ensemble fini d'utilisateurs distincts, définie comme $Topic(U_T) = \{u_1, u_2, ...\}$, où l'élément utilisateur (ou compte) $u_\bullet$ est en outre défini par un 4-tuple d'attributs, $u_\bullet = (UN, SN, UA, TS)$. Chaque attribut dans l'élément utilisateur est défini comme suit:

**Nom d'utilisateur ou *User Name* (UN):** Nous modélisons cet attribut comme un ensemble de caractères ordonnés, définis comme $UN = \{(1, a_1), ..., (i, a_i)\}$, où $i \in \mathbb{Z}_{\geq 0}$, est la position dans la chaîne de nom d'utilisateur et $a_\bullet \in \{Printable\,Characters\}$ [3], est le caractère.

**Nom dans l'écran ou *Screen Name* (SN):** De la même façon que l'attribut *username*, nous modélisons ce champ comme un ensemble ordonné de caractères, défini comme $SN = \{(1, a_1), ..., (i, a_i)\}$ où $i \in \mathbb{Z}_{\geq 0}$ est la position à l'intérieur de la chaîne de noms d'écran, $a_\bullet \in \{Alphanumeric\,Characters\} \cup \{\_\}$, est le caractère.

**Age utilisateur ou *User Age* (UA):** Formellement nous calculons l'âge dans l'unité de temps *days* en soustrayant l'heure courante de la date de création du compte, défini comme $UA = \frac{Time_{now} - Time_{creation}}{864 * 10^5}$, où $Time_{now}, Time_{creation} \in \mathbb{Z}_{\,ge0}$ sont le nombre de millisecondes calculé depuis le 1er janvier 1970, 00:00:00 GMT.

**Ensemble de tweet ou *Tweets Set* (TS):** Chaque utilisateur $u_\bullet \in Topic(U_T)$ peut publier plus d'un tweet dans la rubrique $T$, où chaque tweet peut décrire les pensées ou intérêts de l'utilisateur ou intérêts. Ainsi, nous modélisons ce champ comme un ensemble fini de tweets, définis comme $TS = \{TW_1, ..., TW_n\}$, où $n$ représente le nombre de tweets publiés par l'utilisateur $u_\bullet$. De plus, nous modélisons chaque tweet en double-tuple d'attributs, $TW_\bullet = (Text, Time)$, où $Text = \{(1, w_1), (2, w_2), ...\}$ est un ensemble fini de mots de chaîne, et $Time \in\ mathbbZ_{\geq 0}$ est la date de publication du tweet dans l'unité de temps *minutes* calculée depuis le 1er janvier 1970, 00:00:00 GMT.

**Formalisation du problème.** Etant donné un ensemble d'utilisateurs $U_T$, tels que chaque utilisateur a posté un ou plusieurs tweets dans une thématique tendance $T$, notre problème principal est de prédire le type (spam ou non-spam) de chaque utilisateur dans l'ensemble donné $U_T$, sans nécessiter aucune connaissance préalable telle que la rela-

2. https://support.twitter.com/articles/101125
3. http://web.itu.edu.tr/sgunduz/courses/mikroisl/ascii.html

tion entre les utilisateurs (par exemple les *followers* et les *followees* des utilisateurs). Plus formellement, nous cherchons à concevoir une fonction $y$ qui traite et gère l'ensemble donné d'utilisateurs $U_T$, pour prédire l'étiquette de chaque classe d'utilisateurs, définie comme $y : u_\bullet \rightarrow \{spam, non - spam\}$, $u_\bullet \in U_T$.

### *3.2. Modèle à quatre étapes*

Nous concevons une approche à quatre étapes illustrée par la figure 1 et détaillée ci-dessous.

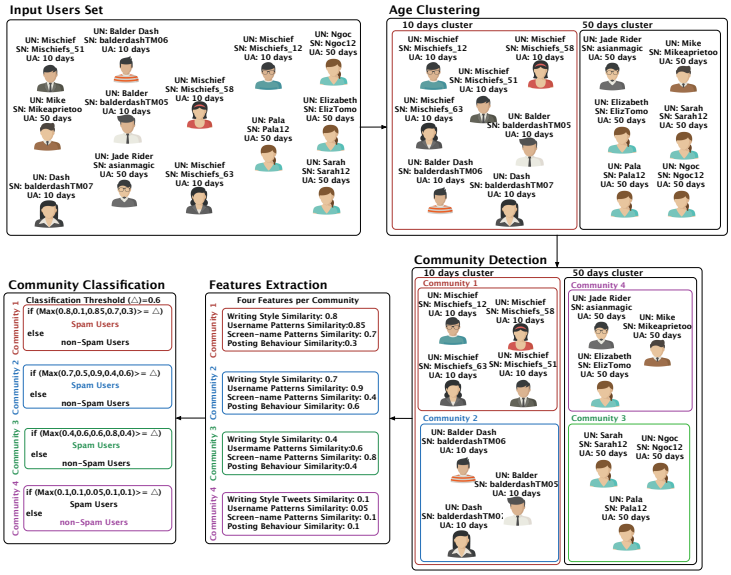


*Figure 1. Un exemple détaillant les quatre étapes du modèle proposé, depuis l'ensemble d'utilisateurs qui a posté des tweets dans une thématique jusqu'à leur classification*

#### *3.2.1. Regroupement en fonction de l'âge des utilisateurs*

Les spammeurs sociaux ont la capacité de créer des centaines et des milliers de comptes Twitter dans un court laps de temps ne dépassant pas quelques jours, pour lancer leurs campagnes de spam. Dans ce cas, la date de création des comptes peut contribuer à regrouper et à isoler les comptes spam qui pourraient avoir une corrélation entre eux. Par conséquent, nous regroupons les utilisateurs en fonction de l'attribut âge de l'utilisateur (UA). D'une manière formelle, soit $C_{age} = \{u|U \in U_T, u.UA = age\}$ un groupe ou *cluster* contenant les utilisateurs ayant un âge égal à $age \in Ages$ où $Ages = \{u.UA|U \in U_T\}$ est un ensemble d'âges d'utilisateurs. De toute évidence, le nombre de groupes d'âges est exactement égal à la taille de $Ages$ (c'est-à-dire $|Ages|$).

#### *3.2.2. Détection de communauté*

Nous exploitons l'utilisation de l'approche de factorisation matricielle non négative ou *non-negative matrix factorization* (NMF) (Yang, Leskovec, 2013), comme une

approche non supervisée, pour déduire la structure des communautés en raison de sa remarquable performance dans le regroupement. Cette approche partitionne une matrice d'information en matrices de facteurs cachés pour un groupe d'âge $C_{age}$, d'utilisateurs, défini mathématiquement comme un problème de minimisation d'optimisation:

$$\min_{H \geq 0} ||\mathbf{X} - \mathbf{H}\mathbf{H}^T||_F^2 \quad (1)$$

Où $|| \bullet ||_F$ est la norme de Frobenius de la matrice considérée, $\mathbf{X} \in R^{|C_{age}| \times |C_{age}|}$ est une matrice d'information représentant la force des connexions sociales entre les utilisateurs, $\mathbf{H} \in R^{|C_{age}| \times K}$ est la matrice des facteurs de la structure communautaire de $K$. L'entrée $X(i,j)$ (i et j représentent les indexes dans la matrice) reflète la force du lien social entre l'utilisateur $u_i \in C_{age}$ et l'utilisateur $u_j \in C_{age}$. L'entrée $H(i,j)$ (i et j représentent les indexes dans la matrice) dans la matrice de facteur caché peut être interprétée comme le degré de confiance de l'utilisateur $u_i \in C_{age}$ appartenant à la communauté $j^{th}$. Il est important de mentionner que chaque utilisateur appartient à une seule communauté uniquement.

Évidemment, inférer la matrice cachée **H** requiert une définition formelle de la matrice d'information **X**. Par exemple, **X** peut être une matrice d'adjacence représentant les liens sociaux ou les liens entre les utilisateurs du groupe d'âge donné $C_{age}$. Cependant, l'obtention de la matrice d'adjacence dans notre cas n'est pas possible car les informations disponibles sur les utilisateurs sont limitées à des méta-données simples décrivant des comptes sans fournir d'informations sur les *followers* et les *followees*. Par conséquent, dans cet article, nous mettons à profit l'information disponible et accessible pour estimer les connexions sociales entre les utilisateurs en proposant deux définitions de la matrice d'information **X** notée $\mathbf{X}^{SN}$ et $\mathbf{X}^{UN}$, dont chacun est formellement défini comme suit:

**Name** ($\mathbf{X}^{SN}$): Comme le champ SN doit être unique, les spammeurs ont tendance à adopter un modèle fixe particulier lors de la création de plusieurs comptes pour agir comme des campagnes de spam. Intuitivement, le chevauchement ou l'appariement élevé dans le SN parmi les utilisateurs augmente la probabilité pour les utilisateurs d'appartenir à la même communauté. Par conséquent, nous définissons la matrice d'information $\mathbf{X}^{SN}$ pour mesurer le degré d'appariement dans l'attribut SN. Plus précisément, étant donné deux utilisateurs $u_i, u_j \in C_{age}$, le degré d'appariement pour une entrée particulière dans la matrice $\mathbf{X}^{SN}$ est défini comme suit:

$$\mathbf{X}^{SN}(i,j) = \frac{max\{|m| : m \in N - gram(u_i.SN) \cap N - gram(u_j.SN), N \in Max\}}{min(|u_i.SN|, |u_j.SN|)}$$

Où $|\bullet|$ est la cardinalité de l'ensemble considéré, $Max = \{1, ..., min(|u_i.SN|, |u_j.SN|)\}$ est un ensemble composé d'entiers positifs représentant le nombre potentiel de caractères qui se chevauchent entre les noms donnés, $N - gram(\bullet)$ est une fonction qui renvoie un ensemble de séquences contiguës de caractères pour le nom donné (ensemble de caractères ordonnés) sur la base de la valeur de $N$. Pour une meilleure compréhension, le 3-gramme (ou le trigramme) d'un SN égale à "vote" est $\{\{(1,v),(2,o),(3,t)\}, \{(1,o),(2,t),(3,e)\}\}$. La définition ci-dessus peut détecter le

motif assorti partout où il apparaît dans l'attribut SN. Par exemple, «vote12» et «tovote» sont des SN pour deux utilisateurs différents de spam, le degré de correspondance selon l'équation 3.2.2 est autour de $(\frac{4}{6})66.6\%$, résultant de l'utilisation du motif «vote», quelle que soit la position du motif.

**Similarité de nom d'utilisateur** ($\mathbf{X}^{UN}$): Contrairement à l'attribut SN, les spammeurs peuvent copier l'attribut de nom d'utilisateur autant qu'ils le souhaitent. Ils exploitent des noms représentatifs (pas aléatoires) pour attirer les utilisateurs non-spam. Par conséquent, la correspondance totale ou partielle entre les utilisateurs dans cet attribut augmente les performances de détection de la communauté. Nous définissons la matrice d'information $\mathbf{X}^{UN}$ pour mesurer le degré de similarité entre les utilisateurs dans l'attribut de nom d'utilisateur. Formellement, étant donné deux utilisateurs $u_i, u_j \in C_{age}$, le degré de similarité est défini comme suit:

$$\mathbf{X}^{UN}(i,j) = \frac{max\{|m| : m \in N-gram(u_i.UN) \cap N-gram(u_j.UN), N \in Max\}}{min(|u_i.UN|, |u_j.UN|)} \tag{2}$$

Où $Max = \{1, ..., min(|u_i.UN|, |u_j.UN|)\}$.

**Combiner les informations des matrices.** Avec ces deux matrices d'information, l'approche NMF permet de les intégrer dans la même fonction objective. Ainsi, la nouvelle version de la fonction objectif est définie comme suit:

$$\min_{H \geq 0} ||\mathbf{X}^{SN} - \mathbf{H}\mathbf{H}^T||_F^2 + ||\mathbf{X}^{UN} - \mathbf{H}\mathbf{H}^T||_F^2 \tag{3}$$

De toute évidence, l'équation 3 infère la matrice de facteurs cachés $H$ pour représenter la structure communautaire cohérente des utilisateurs.

**Approche d'optimisation.** La fonction objective n'est pas conjointement convexe et aucune solution de forme fermée existe. Par conséquent, nous proposons l'utilisation de la descente en gradient comme une autre approche d'optimisation (Yang, Leskovec, 2013). Comme nous avons une variable libre de matrice (**H**), l'approche de descente par gradient la met à jour itérativement jusqu'à ce que la variable converge. Nous exploitons deux conditions d'arrêt: (i) le nombre d'itérations, noté $M$; (ii) et la variation absolue de la matrice $H$ dans deux itérations consécutives est inférieure à un seuil, c'est-à- dire $|(||H^{\tau}||_F - ||H^{\tau-1}||_F)| \leq \epsilon$.

### *3.2.3. Extraction des caractéristiques basées sur la communauté*

Afin de classer chaque communauté, un ensemble de caractéristiques peut être extrait pour discriminer parmi les communautés spam ou non-spam. Aucune caractéristique n'est capable de discriminer efficacement entre les communautés de spam et non-spam. De plus, la conception des caractéristiques doit maintenir la condition que la récupération d'informations sur les utilisateurs de serveurs Twitter n'est pas autorisée du tout. Par conséquent, nous proposons une conception de quatre caractéristiques qui examinent la perspective collective des utilisateurs, en utilisant uniquement les informations disponibles sur les utilisateurs. Nos caractéristiques sont réparties entre les caractéristiques basées sur le tweet et celles basées sur l'utilisateur,

répertoriées comme: *username patterns similarity* (**UNPS**), *screen-name patterns similarity* (**SNPS**), *tweets writing style similarity* (**TsWSS**) et *tweets posting behavior similarity* (**TPBS**). Nous formalisons la $j^{eme}$ communauté extraite comme sextuple d'attributs, $C_j = \{U, UNPS, SNPS, TsWSS, TPBS, L\}$ où $U$ est un ensemble d'utilisateurs appartenant à la $j^{eme}$ communauté qui peut être extraite de **H** matrix, et $L \in \{spam, non-spam\}$ est l'étiquette de la $j^{eme}$ communauté.

**UNPS et SNPS**: Les spammeurs peuvent adopter un modèle pour créer leurs campagnes de spam. Ainsi, lorsqu'une communauté est biaisée par rapport à un modèle particulier utilisé dans la création de comptes, cette communauté a une forte probabilité d'être une campagne de spam et, par conséquent, tous les utilisateurs de cette communauté sont des utilisateurs spam (comptes). Nous modélisons ce comportement de spam en trouvant d'abord tous les modèles possibles utilisés pour nommer des comptes, extraits des attributs UN et SN. Ensuite, nous calculons la distribution de probabilité de motifs extraits de UN ou SN. Enfin, nous comparons la distribution de probabilité calculée d'un attribut avec la distribution uniforme des motifs extraits. En effet, nous émettons l'hypothèse que la distribution de probabilité des modèles de communautés non-spam doit être proche de la distribution uniforme.

De manière formelle, soit $PT^{UN}$ et $PT^{SN}$ deux ensembles de chaînes de caractères finis extraits des attributs UN et SN de la $j^{eme}$ communauté, respectivement. De même, soit $P_D^{UN}$ et $P_D^{SN}$ les distributions de probabilité des patrons d'attributs UN et SN, respectivement. Pour la distribution uniforme, soit $P_{uniforme}$ une distribution uniforme correspondante des modèles d'attributs considérés. Par exemple, dans le cas d'une communauté donnée, notons $PT^{SN} = \{"mischief", "isch", "\_12", "\_14"\}$, $P_D^{SN} = \{("mischief",\ 0.7), ("\_15", 0.1), ("\_14", 0.1), ("\_12", 0.1)\}$ un ensemble de motifs de SN avec sa distribution de probabilité uniforme. La distribution de probabilité de ces modèles est $\{("mischief", 0.25), ("\_15", 0.25), ("\_14", 0.25), ("\_12", 0.25)\}$.

Pour extraire des chaînes de caractères à partir d'un UN ou d'un SN, nous appliquons l'approche des caractères N-gram sur le UN ou le SN dans la communauté inférée. Comme les spammeurs peuvent définir des modèles variant en longueur et en position, pour prendre tous ou la plupart des modèles possibles, nous utilisons différentes valeurs de $N$ allant de trois à la longueur du nom. Nous évitons $N \in \{1, 2\}$ car il est inutile d'avoir un ou deux caractères de modèle. Formalement, pour un nom donné sous la forme de $Name = \{(1, a_1), (2, a_2), ...\}$, dont les profils potentiels sont extraits par:

$$Patterns(Name) = \bigcup_{N \in Max} N - gram(Name) \tag{4}$$

Où $Max = \{3, ..., \mid Name \mid\}$ est un ensemble fini de valeurs possibles de N.

Avec la définition introduite, les ensembles de modèles de chaînes de la $j^{eme}$ communauté sont donnés comme suit:

$$PT^{UN} = \bigcup_{u \in c_j \cdot U} Patterns(u \cdot UN), \quad PT^{SN} = \bigcup_{u \in c_j \cdot U} Patterns(u \cdot SN) \tag{5}$$

Nous quantifions la similarité entre les distributions en effectuant une corrélation croisée entre la distribution de probabilité des modèles associés à une communauté et la distribution uniforme correspondante aux modèles considérés. Formalement, nous calculons la similarité de distribution de probabilité pour les attributs UN et SN à l'aide des formules suivantes:

$$UNPS(C_j) = 1 - \frac{Area(P_D^{UN} \star P_{uniform})}{Area(P_{uniform} \star P_{uniform})} \tag{6}$$

$$SNPS(C_j) = 1 - \frac{Area(P_D^{SN} \star P_{uniform})}{Area(P_{uniform} \star P_{uniform})} \tag{7}$$

Où $Area\,(\bullet)$ est une fonction qui calcule la zone sous la nouvelle distribution résultante par l'opération de corrélation. L'idée-clé d'effectuer l'auto-corrélation entre la distribution de probabilité uniforme et elle-même, est de normaliser la valeur de la zone qui provient de l'opération de corrélation croisée en rangeant les caractéristiques entre zéro et 1. Les valeurs de ces deux caractéristiques ont une corrélation directe avec la probabilité d'être une communauté de spam.

**TsWSS**: Chaque communauté est inférée par un ensemble de tweets posté par ses utilisateurs. Étant donné que les spammeurs automatisent leurs campagnes de spam, la probabilité de trouver une corrélation dans le style d'écriture parmi les tweets considérés est élevée. Par exemple, les tweets spam d'une campagne ont une structure de style commune pour écrire des tweets (mot, mot, hashtag, mot, mot, mot, mot, mot et après URL). Nous modélisons cette caractéristique en transformant d'abord les textes des tweets en un niveau supérieur d'abstraction. Ensuite, nous mesurons la similarité du style d'écriture parmi les tweets de la $j^{eme}$ communauté en utilisant la similarité de Jaccard. Une fonction de transformation $Type(ST) \in \{W, H, U, M\}$, est définie qui prend une chaîne de caractère $ST$ comme paramètre et renvoie le type de la chaîne d'entrée (**W**ord, **H**ashtag, **U**rl et **M**ention). Par conséquent, pour un tweet $TW_\bullet$, le nouvel ensemble de représentation d'abstraction est $Trans(TW_\bullet) = \{(i, Type(S))|(I, S) \in TW_\bullet.Text\}$ où $i$ est la position de la chaîne de mots dans le texte du tweet et $S$ est une chaîne de mots qui nécessite une transformation. Avec ces définitions, nous calculons la similarité de style d'écriture parmi un ensemble de tweets comme suit:

$$TsWSS(C_j) = \frac{\sum_{T_1 \in Tweets_j} \sum_{T_2 \in Tweets_j} \frac{|Trans(T_1) \cap Trans(T_2)|}{|Trans(T_1) \cup Trans(T_2)|}}{(|Tweets_j|)(|Tweets_j| - 1)} \tag{8}$$

où $Tweets_j = \bigcup_{u \in C_j} u.TS$ est une unification de tous les tweets affichés par les utilisateurs de la $j^{eme}$ communauté inférée. Les valeurs supérieures et inférieures de cette fonction sont un et zéro respectivement. La valeur élevée de cette fonction signifie que la probabilité de la $j^{eme}$ communauté d'être un spam est élevée en raison de la proximité des tweets dans le style d'écriture.

**TPBS**: Le comportement de partage (*posting behaviour*) (par exemple, toutes les 5 min) de tweets au niveau de synchronisation pourrait être un indice important supplémentaire dans l'identification des communautés de spam. Ainsi, nous proposons une caractéristique qui mesure la corrélation entre le comportement de partage des utilisateurs. Nous modélisons ce comportement en calculant d'abord la distribution de probabilité de temps d'affichage de chaque utilisateur appartenant à une communauté particulière.

Pour chaque couple d'utilisateurs, nous mesurons la similarité de leurs distributions de probabilité de temps d'affichage en utilisant le concept de corrélation croisée, ce qui donne une valeur réelle unique comprise entre 0 et 1. Ensuite, nous calculons la distribution de probabilité de la similarité de temps d'affichage pour la comparer avec une distribution uniforme tracée sur l'intervalle [0,1]. Plus formellement, soit $P^u_{TS}$ une distribution de probabilités de temps d'affichage des tweets de l'utilisateur $u$. $P^u_{TS}$ peut être tiré simplement à partir du temps tweets de l'utilisateur $u$ qui les a déjà affichés dans la rubrique $T$. Nous calculons la similarité entre deux distributions de temps d'affichage de deux utilisateurs différents appartenant à $u_1, u_2 \in C^{j^{eme}}$ communauté, comme suit:

$$PostSim(u_1, u_2) = \frac{Area(P^{u_1}_{TS} \star P^{u_2}_{TS})}{Min(Area(P^{u_1}_{TS} \star P^{u_1}_{TS}), Area(P^{u_2}_{TS} \star P^{u_2}_{TS}))} \quad (9)$$

Où $Area(\bullet)$ est une fonction qui calcule la zone sous la nouvelle distribution résultante, $Min(\bullet, \bullet)$ est une opération minimale qui sélectionne la zone minimale. Le point clé de la prise de l'opération min est de normaliser la zone qui résulte de la corrélation croisée. La valeur faible de $PostSim$ signifie que les deux utilisateurs d'entrée ont une faible corrélation dans le comportement de temps d'affichage.

En calculant la valeur finale de la fonction $TPBS$, nous calculons d'abord la distribution de probabilité de $PostSim$ sur toutes les paires d'utilisateurs possibles existant dans la $C^{j^{eme}}$ communauté. Soit $P_{PostSim}$ (par exemple $\{(0.25, 0.4), (0.1, 0.6)\}$) la distribution de probabilités de la similarité d'écriture et $P^{Uniform}_{PostSim}$ (par exemple $\{(0.25, 0.5), (0.1, 0.5)\}$) la distribution uniforme correspondante de $PostSim$. Nous quantifions les différences entre les distributions en effectuant une corrélation croisée entre elles, définie comme suit:

$$TPBS(C_j) = 1 - \frac{Area(P_{PostSim} \star P^{Uniform}_{PostSim})}{Area(P^{Uniform}_{PostSim} \star P^{Uniform}_{PostSim})} \quad (10)$$

Où la valeur élevée (près de 1) de $TPBS$ signifie que tous les utilisateurs de la $j^{eme}$ communauté ont presque le même comportement positif (c'est-à-dire presque la même fréquence d'affichage) et que cette communauté a une forte probabilité d'être une campagne de spam. D'une autre côté, lorsque le $P_{PostSim}$ est proche de la distribution uniforme, cela signifie que presque aucun utilisateur n'a le même comportement de publication et donc que la communauté a une faible probabilité d'être une campagne de spam.

### *3.2.4. Fonction de Classification*

Nous utilisons les quatre caractéristiques proposées basées sur la communauté pour prédire l'étiquette de classe de chaque utilisateur qui partage des tweets dans la rubrique $T$. Nous classons les utilisateurs d'une communauté en spam si et seulement si l'une des quatre caractéristiques est supérieure à un seuil donné. L'intuition derrière cette proposition est que les caractéristiques conçues sont de détecter les communautés de spam, ce qui signifie que d'avoir au moins une valeur de caractéristique élevée est suffisant pour juger si une communauté est spam. Alors que, pour juger une communauté comme non-spam, il est obligatoire de s'assurer que toutes les caractéristiques sont de faible valeur. Par conséquent, nous concevons la fonction de classification finale, $y$, de sorte qu'elle

attribue un label à un utilisateur d'entrée $u \in C_j$ basé sur les quatre caractéristiques communautaires, définies comme suit:

$$\text{y(u)} = \begin{cases} spam & u \in C_j \;\&\; (TsWss(C_j) \geq \Delta || \; TPBS(C_j) \geq \Delta \\ & \quad || \; SNPS(C_j) \geq \Delta || \; UNPS(C_j) \geq \Delta) \\ non-spam & u \in C_j \;\&\; TsWss(C_j) < \Delta \& \; TPBS(C_j) < \Delta \\ & \quad \& \; SNPS(C_j) < \Delta \& \; UNPS(C_j) < \Delta \end{cases} \tag{11}$$

Où $\Delta$ est un seuil de classification déterminé en fonction de la métrique de performance (par exemple exactitude, rappel, précision) qui doit être optimisée.

## 4. Description de la base de données et de la vérité terrain

**Méthode d'aspiration (*Crawling)*.** Nous exploitons notre *crawler* de l'équipe de recherche pour collecter des comptes et des tweets, lancés depuis le 1/Jan/2015. l'approche de diffusion en continu est utilisée pour obtenir un accès pour 1% des tweets globaux, en tant qu'approche d'analyse impartiale. Une telle approche est couramment exploitée dans la littérature pour recueillir et créer des données dans les recherches sur les OSN.

*Tableau 1. Statistiques des utlisateurs (comptes) et des tweets annotés*

| | Spam | non-Spam |
|---|---|---|
| Number of Tweets | 763,555 (11.8%) | 5,707,254 (88,2%) |
| Number of Users | 185,843(8.9%) | 1,902,288(91.1%) |

**Description de la base de données.** En utilisant notre ensemble de données Twitter de l'équipe, nous avons regroupé les tweets collectés en fonction de la thématique disponible dans le tweet en ignorant les tweets qui ne contiennent aucune thématique. Ensuite, nous avons sélectionné les tweets de 100 thématiques tendances (par exemple #Trump) échantillonnés au hasard pour mener nos expérimentations. Nous exploitons de cette façon l'exploration et l'échantillonnage pour éliminer toute polarisation possible dans les données et tirer des conclusions impartiales.

**Base de données vérité terrain.** Pour évaluer l'efficacité des caractériqtiques (*patterns)* décrivant le spam dans la récupération des comptes spam, nous avons créé un ensemble de données annotées en étiquetant chaque compte comme spam ou non-spam. Cependant, avec l'énorme quantité de comptes, l'utilisation de l'approche par annotation manuelle pour avoir des jeux de données étiquetés est une solution peu pratique. Par conséquent, nous exploitons un processus d'annotation largement suivi dans les recherches de détection de spam social. Le processus vérifie si l'utilisateur de chaque tweet a été suspendu par Twitter. En cas de suspension, l'utilisateur avec ses tweets sont étiquetés comme spam; sinon nous assignons non-spam pour les deux. Au total, comme indiqué dans le Tableau 1, nous avons constaté que plus de 760 000 tweets ont été classés comme spam, génèrés par près de 185 800 comptes spam.

## 5. Résultats et Evaluations

Nous présentons dans cette section, les résultats obtenus en comparant notre approche avec d'autres algorithmes de la littérature.

### *5.1. Configuration expérimentale*

**Métriques de précision.** Comme la vérité terrain de chaque classe d'étiquette de chaque tweet est donnée, nous utilisons l'exactitude (*accuracy)*, la précision, le rappel, la F-mesure, la précision moyenne, le rappel moyen et la F-mesure moyenne; calculée en fonction de la matrice de confusion de l'outil Weka (Hall *et al.*, 2009); comme métriques couramment utilisées dans les problèmes de classification. Comme notre problème est la classification en deux classes (binaires), nous calculons la précision, le rappel et la F-mesure pour la classe «spam», alors que les métriques de moyennes combinent les deux classes en fonction de la fraction de chaque classe (par exemple 11,8 % * "Précision de spam" + 88,2 % * "précision de non-spam").

***Baselines* ou données de référence.** Nous définissons deux baselines pour comparer notre approche avec eux, à savoir: (i) baseline "A" qui représente les résultats lors de la classification de tous les tweets comme non-spam directement sans classement; (ii) baseline "B" qui montre les résultats obtenus lors de l'application d'algorithmes d'apprentissage supervisés selon les caractéristiques associées au "tweet" . Comme de nombreux algorithmes d'apprentissage fournis par l'outil Weka, nous exploitons *Naive Bayes*, *Random Forest*, *J48,* et *Support Vector Machine* (SVM) comme approches d'apprentissage supervisé connues pour évaluer la performance des caractéristiques mentionnées.

*Tableau 2. Résultats de performance du baseline A et du baseline B selon différents paramètres.*

| Learning Algorithm | Accuracy | Precision | Recall | F-Measure | Avg. Precision | Avg.Recall | Avg. F-Measure |
|---|---|---|---|---|---|---|---|
| **Baseline (A): All Tweets Labeled as Non-Spam** | | | | | | | |
| ———— | 91.1% | 0.0% | 0.0% | 0.0.% | 91.1% | 91.1% | 91.1% |
| **Baseline (B): Supervised Machine Learning Approach** | | | | | | | |
| Naive Bayes | 81.2% | 13.7% | 10.5% | 11.9% | 79.0% | 81.2% | 80.1% |
| Random Forest (#Trees=100) | 86.4% | 13.2% | 2.8% | 4.6% | 79.0% | 86.4% | 80.1% |
| Random Forest (#Trees=500) | 86.5% | 12.6% | 2.6% | 4.7% | 79.4% | 86.5% | 82.8% |
| J48 (Confidence Factor=0.2 ) | 86.4% | 13.8% | 2.9% | 4.9% | 79.6% | 86.4% | 82.5% |
| SVM (Gamma=0.5) | 87.2% | 15.7% | 0.2% | 0.4% | 78.3% | 87.2% | 82.5% |
| SVM (Gamma=1.0) | 87.0% | 15.9% | 0.1% | 0.3% | 77.9% | 87.0% | 82.2% |

**Paramétrage des *Baselines*.** Pour l'approche *Naive Bayes*, nous définissons les options "*useKernelEstimator*" et "*useSupervisedDiscretization*" à false comme valeurs par défaut définies par Weka. Pour *Random Forest*, nous avons mis l'option "*max depth*" à 0 (illimité), en étudiant l'effet du changement du nombre d'arbres $\in \{100, 500\}$. Pour l'approche $J48$, nous fixons le nombre minimum d'instances par feuille à 2, le nombre de plis à 3 et le facteur de confiance à 2. Pour l'approche SVM, nous utilisons l'implémentation *LibSVM* (Chang, Lin, 2011) intégrée à l'outil Weka pour définir la fonction du noyau sur *Radial Basis* et examiner l'impact de gamma $\in \{0.5, 1\}$, où les paramètres restants sont par défaut.

**Paramétrage de notre approche.** Pour l'étape de détection communautaire, nous fixons $\eta$ = 0.001, $M$ = 10,000 et $\epsilon = 0.0001$ comme valeurs pour le taux d'apprentis-

sage, le nombre d'itérations et le seuil de changement absolu dans la matrice cachée **H**, respectivement. Pour le nombre de communautés $K$, nous expérimentons notre approche à deux valeurs différentes, $K \in \{5, 10\}$, pour en étudier l'effet. Pour la taille des matrices d'information **X**, nous considérons tous les utilisateurs distincts (comptes) de chaque hashtag sans exclure aucun utilisateur disponible dans la collection d'essai. Comme un algorithme itératif est utilisé pour résoudre le problème d'optimisation de la détection communautaire, nous initialisons chaque entrée de la matrice cachée **H** par une petite valeur réelle positive tirée d'une distribution uniforme sur l'intervalle $[0, 1]$. Pour le seuil $\Delta$, nous étudions l'impact de la modification de sa valeur en effectuant des expérimentations à différentes valeurs de $\Delta \in [0.1, 1.0]$ avec un pas d'incrémentation de $0.1$.

### *5.2. Résultats Expérimentaux*

**Résultats de *Baselines*.** Selon les résultats des baselines rapportées dans le Tableau 2, les modèles de classification supervisés ont une forte défaillance dans le filtrage des tweets spam existant dans les 100 thématiques tendances. Cet échec peut être facilement identifié à partir des valeurs basses de rappel de spam ($4^{eme}$ colonne) où la valeur la plus élevée est obtenue par l'algorithme d'apprentissage $NaiveBayes$. Le 10,5 % du rappel de spam obtenu par $NaiveBayes$ signifie que moins de 80 000 tweets spam peuvent être détectés à partir de 736 500 tweets spam. Les faibles valeurs de précision du spam indiquent également qu'un nombre important de tweets «non-spam» a été classé en «spam». Par la suite, comme la F-mesure de spam dépend des mesures de rappel et de précision, les valeurs de la F-mesure de spam sont évidemment faibles. Les valeurs de précision du baseline «B» sont proches des valeurs de précision de la baseline «A». Toutefois, compte tenu des faibles valeurs de précision du spam et du rappel de spam, la métrique d'exactitude dans ce cas n'est pas une mesure indicative et utile pour juger l'apprentissage supervisé comme une approche efficace. Plus précisément, l'approche d'apprentissage supervisé n'ajoute pas une contribution significative à l'amélioration de la qualité des 100 tweets des thématiques tendances. L'idée clé de l'utilisation de différents algorithmes d'apprentissage est de varier leurs paramètres est de mettre en évidence la mauvaise qualité des techniques de l'état de l'art traitant le tweet. Dans l'ensemble, les résultats obtenus par les modèles permettent de tirer diverses conclusions: (i) les techniques de l'état de l'art ne sont pas discriminatives entre les tweets non-spam et spam, assurant la dynamique du contenu spam; (ii) les spammeurs ont tendance à publier des tweets presque similaires aux non-spam; (iii) l'adoption d'une approche supervisée pour effectuer l'apprentissage sur un ensemble de données annotées de thématiques tendances et l'application du modèle de classification sur des thématiques tendances futures ou non annotées n'est *pas* la solution du tout.

**Effet de $\Delta$.** En examinant les performances de notre approche dans le Tableau 3, le comportement est complètement différent lors du rappel (classement) des comptes spam, surtout lorsque la valeur de $\Delta$ devient plus faible. Les résultats de rappel sont tout à fait compatibles avec l'équation 11 conçue pour classer les utilisateurs. Pour des valeurs faibles de $\Delta$, certaines communautés non-spam sont classées comme spam. En effet, cela explique la dégradation dramatique de la précision lors de la diminution de

*Tableau 3. Les résultats de performance de notre approche en plusieurs métriques, calculées à différentes valeurs de nombre de communautés $K \in \{5, 10\}$ et à divers seuils de classification $\Delta \in \{0.1, 1\}$.*

| Model ($\Delta$) | Accuracy | Spam Precision | Spam Recall | Spam F-measure | Avg. Precision | Avg. Recall | Avg. F-measure |
|---|---|---|---|---|---|---|---|
| **Number of Communities** ($K = 5$) | | | | | | | |
| $\Delta$=0.1 | 63.0% | 14.6% | **65.8%** | 23.9% | **87.9%** | 63.0% | 73.4% |
| $\Delta$=0.2 | 66.3% | 15.4% | 63.0% | 24.8% | 87.8% | 66.3% | 75.6% |
| $\Delta$=0.3 | 68.1% | 15.9% | 60.7% | **25.2%** | 87.8% | 68.1% | 76.7% |
| $\Delta$=0.4 | 69.7% | 16.1% | 57.5% | 25.1% | 87.5% | 69.7% | 76.7% |
| $\Delta$=0.5 | 77.4% | 18.2% | 44.3% | 25.8% | 87.0% | 77.4% | 81.9% |
| $\Delta$=0.6 | 78.2% | 17.7% | 40.3% | 24.6% | 86.7% | 78.2% | 82.2% |
| $\Delta$=0.7 | 82.0% | 18.6% | 30.9% | 23.3% | 86.3% | 82.0% | 84.0% |
| $\Delta$=0.8 | 85.6% | 19.7% | 20.2% | 19.9% | 85.8% | 85.6% | 85.7% |
| $\Delta$=0.9 | 89.1% | **20.3%** | 7.6% | 11.1% | 85.2% | 89.1% | **87.1%** |
| $\Delta$=1.0 | **91.1%** | 0.0% | 0.0% | 0.0% | 83.1% | **91.1%** | 86.9% |
| **Number of Communities** ($K = 10$) | | | | | | | |
| $\Delta$=0.1 | 69.0% | 15.9% | **58.6%** | 25.1% | 87.6% | 69.0% | 77.2% |
| $\Delta$=0.2 | 71.5% | 16.7% | 56.0% | 25.8% | **87.6%** | 71.5% | 78.8% |
| $\Delta$=0.3 | 73.3% | 17.2% | 53.3% | **26.0%** | 87.5% | 73.3% | 79.7% |
| $\Delta$=0.4 | 75.1% | 17.5% | 49.0% | 25.8% | 87.2% | 75.1% | 80.7% |
| $\Delta$=0.5 | 81.3% | 19.5% | 35.7% | 25.2% | 86.7% | 81.3% | 83.9% |
| $\Delta$=0.6 | 82.1% | 19.0% | 31.5% | 23.7% | 86.3% | 82.1% | 84.2% |
| $\Delta$=0.7 | 85.1% | 19.8% | 22.3% | 21.0% | 85.9% | 85.1% | 85.5% |
| $\Delta$=0.8 | 87.7% | **20.7%** | 13.5% | 16.3% | 85.6% | 87.7% | 86.6% |
| $\Delta$=0.9 | 90.0% | 20.7% | 4.0% | 6.6% | 85.0% | 90.0% | **87.4%** |
| $\Delta$=1.0 | **91.1%** | 0.0% | 0.0% | 0.0% | 83.1% | **91.1%** | 86.9% |

la valeur de $\Delta$, ainsi que la dégradation de la précision du spam. Bien que les valeurs de rappel sont élevées, les valeurs de précision de spam de notre approche sont presque semblables à celles de l'approche d'apprentissage supervisé.

**Impact de $K$.** Le nombre de communautés $K$, a un impact direct et évident sur l'exactitude (*accuracy)*, la précision et les mesures de rappel de spam. En effet, l'exactitude et la précision du spam augmentent tant que le nombre de communautés augmente. La justification de ce comportement est que les 100 thématiques tendances expérimentés ont été attaqués par de nombreuses campagnes de spam non corrélées. Par la suite, l'augmentation du nombre de communautés permet de détecter les campagnes de spam de manière précise et exacte. Au niveau du rappel de spam, le comportement est inversement corrélé au nombre de communautés. L'utilisation d'un plus grand nombre de communautés que le nombre réel (inconnu) de campagnes de spam conduit à séparer ces campagnes sur plus de communautés. Comme les communautés de spam peuvent contenir des utilisateurs non-spam (comptes); dans ce cas, les valeurs des caractéristiques diminuent, classifiant ces communautés comme "non-spam".

## 6. Conclusion

Dans cet article, nous avons décrit une approche non supervisée pour filtrer les utilisateurs spam dans les collections à grande échelle de thématiques "tendances". Notre approche adopte la perspective collective dans la détection des utilisateurs spammeurs en découvrant les corrélations entre eux. Notre travail apporte deux contributions au domaine de la qualité de l'information: (i) le filtrage des utilisateurs sans avoir besoin de jeux de données annotés; (ii) un processus de filtrage rapide en raison de la dépendance des méta-données disponibles, sans avoir recours à l'information des serveurs de Twitter. Avec cette nouvelle idée, nous planifions d'étudier l'impact de la collaboration avec d'autres OSN pour améliorer les résultats actuels.

*Remerciements*

*Ce travail s'intègre dans le cadre des contributions du projet ANR FILTER 2.*

## Bibliographie

Abascal-Mena R., Lema R., Sèdes F. (2015). Detecting sociosemantic communities by applying social network analysis in tweets. *Social Netw. Analys. Mining*, vol. 5, n° 1, p. 38:1–38:17. Consulté sur http://dx.doi.org/10.1007/s13278-015-0280-2

Benevenuto F., Magno G., Rodrigues T., Almeida V. (2010). Detecting spammers on twitter. In *In collaboration, electronic messaging, anti-abuse and spam conference (ceas)*, p. 12.

Canut C. M., On-at S., Péninou A., Sèdes F. (2015). Time-aware egocentric network-based user profiling. In *Proceedings of the 2015 IEEE/ACM international conference on advances in social networks analysis and mining, ASONAM 2015, paris, france, august 25 - 28, 2015*, p. 569–572. Consulté sur http://doi.acm.org/10.1145/2808797.2809415

Cao C., Caverlee J. (2015). Detecting spam urls in social media via behavioral analysis. In *Advances in information retrieval*, p. 703–714. Springer.

Chang C.-C., Lin C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, vol. 2, p. 27:1–27:27. (Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm)

Chu Z., Gianvecchio S., Wang H., Jajodia S. (2012). Detecting automation of twitter accounts: Are you a human, bot, or cyborg? *Dependable and Secure Computing, IEEE Transactions on*, vol. 9, n° 6, p. 811–824.

Chu Z., Widjaja I., Wang H. (2012). Detecting social spam campaigns on twitter. In *Applied cryptography and network security*, p. 455–472.

Hall M., Frank E., Holmes G., Pfahringer B., Reutemann P., Witten I. H. (2009, novembre). The weka data mining software: An update. *SIGKDD Explor. Newsl.*, vol. 11, n° 1, p. 10–18.

Martinez-Romo J., Araujo L. (2013). Detecting malicious tweets in trending topics using a statistical analysis of language. *Expert Systems with Applications*, vol. 40, n° 8, p. 2992–3000.

McCord M., Chuah M. (2011). Spam detection on twitter using traditional classifiers. In *Proceedings of the 8th international conference on autonomic and trusted computing*, p. 175–186. Springer-Verlag.

Stringhini G., Kruegel C., Vigna G. (2010). Detecting spammers on social networks. In *Proceedings of the 26th annual computer security applications conference*, p. 1–9. New York, NY, USA, ACM.

Wang A. H. (2010, July). Don't follow me: Spam detection in twitter. In *Security and cryptography (secrypt), proceedings of the 2010 international conference on*, p. 1-10.

Yang J., Leskovec J. (2013). Overlapping community detection at scale: A nonnegative matrix factorization approach. In *Proceedings of the sixth acm international conference on web search and data mining*, p. 587–596. New York, NY, USA, ACM. Consulté sur http://doi.acm.org/10.1145/2433396.2433471