Revista de Sistemas de Informação da FSMA
n. 13 (2014) pp. 21-29

SISTEMAS DE INFORMAÇÃO
http://www.fsma.edu.br/si/sistemas.html

TRILHA PRINCIPAL

Avaliação da Estabilidade e Robustez de Algoritmos para Recomendação de Serviços Web Semânticos Submetidos a Ataques de Injeção de Perfis

Pedro Henrique Grandin e Juan Manuel Adán-Coello

***Resumo* — Sistemas de recomendação baseados em filtragem colaborativa são abertos por natureza, o que os torna vulneráveis a ataques de injeção de perfis que procuram inserir avaliações tendenciosas na base de dados do sistema, possibilitando que atacantes possam manipular as suas recomendações. Neste artigo avalia-se a estabilidade e a robustez de algoritmos de filtragem colaborativa aplicados à recomendação de serviços Web semânticos quando submetidos a ataques de injeção de perfis dos tipos aleatório e segmento. Foram avaliados quatro algoritmos: (1) o IMEAN, que faz previsões usando a média das avaliações recebidas pelo item alvo, (2) o UMEAN, que faz previsões usando a média das avaliações feitas pelo usuário alvo; (3) algoritmo baseado no método dos k vizinhos mais próximos (*k-nn*) e (4) algoritmo baseado no método de agrupamento *k-médias*. Os experimentos realizados mostraram que o algoritmo UMEAN não é afetado pelos ataques e que o IMEAN é o mais vulnerável de todos. Entretanto, o UMEAN e o IMEAN têm pouco potencial de aplicação prática devido à baixa precisão de suas previsões. Entre os algoritmos com tolerância intermediária aos ataques, mas com bom desempenho preditivo, o *k-nn* apresentou-se mais robusto e estável que o *k-médias*.**

***Palavras-chave* — Ataque de injeção de perfis, algoritmos de filtragem colaborativa, serviços Web semânticos.**

I. INTRODUÇÃO

NAS arquiteturas SOA (*Service Oriented Architecture*), serviços fracamente acoplados permitem a criação de processos de negócios flexíveis e dinâmicos e aplicações ágeis que podem envolver diversas organizações e plataformas computacionais [1]. Entre os problemas centrais à criação desses processos, fruto da composição de novos serviços a partir dos existentes, está a descoberta de serviços que atendam aos requisitos e interesses dos usuários, sejam eles pessoas ou agentes de software.

Pedro Henrique Grandin (e-mail: pedro.hg@puccamp.edu.br) e Juan Manuel Adán-Coello (juan@puc-campinas.edu.br) estão vinculados à Faculdade de Engenharia de Computação da Pontifícia Universidade Católica de Campinas (PUC-Campinas).

A descoberta de um serviço é feita por algoritmos de casamento (*matching*) que buscam em um repositório de descrições quais dos serviços anunciados atendem aos requisitos do potencial usuário. As arquiteturas de descoberta de serviços, em geral baseadas nos padrões WSDL [2] e UDDI [3], apresentam uma séria de limitações que dificultam a localização de serviços relevantes para os usuários. Tais limitações devem-se, em grande parte, ao uso de descrições informais da funcionalidade e capacidades dos serviços, feitas em língua natural, em geral sem a utilização de um vocabulário que seja comum ao requisitante e ao provedor do serviço. Os Serviços Web Semânticos são uma abordagem recente que procura superar essas limitações combinando a tecnologia de serviços Web com elementos da denominada Web Semântica [4] [5].

As pesquisas na área de serviços Web têm focado nos algoritmos de casamento de serviços, semânticos ou não. Entretanto, trabalhos recentes evidenciam que a questão mais desafiadora quando se quer fornecer ao usuário o serviço desejado é a seleção do serviço a partir de uma lista de candidatos e não o processo de casamento em si. As principais abordagens para a seleção de serviços incluem a filtragem baseada em conteúdo, a filtragem colaborativa, sistemas de reputação, sistemas P2P, sistemas de referências [6]. Entre eles, os métodos de filtragem colaborativa, foco deste artigo, estão entre os mais relevantes.

A filtragem colaborativa baseia-se na suposição de que usuários com perfis, ou interesses, comuns normalmente procuram itens similares. A idéia fundamental é tomar como base para recomendar itens a um usuário as opções de usuários semelhantes quando escolheram itens relacionados [7].

O problema da recomendação pode ser reduzido a estimar avaliações para itens que não foram acessados por um usuário e recomendar-lhe aqueles com as maiores avaliações estimadas.

Embora sistemas de recomendação possam ser vistos como parte de um sistema de busca, eles podem também ser usados de forma autônoma a fim de sugerir itens aos usuários à

medida que estes se tornem disponíveis, sem a necessidade de consulta explícitas e constantes.

A filtragem colaborativa é largamente usada em sistemas disponíveis na Web para recomendar diversos tipos de itens, incluindo livros, músicas e filmes.

O desenvolvimento de novos algoritmos que aumentem a precisão e eficácia de sistemas de recomendação ou produzam modelos que expliquem as razões por trás de uma recomendação é tema de crescente interesse acadêmico e comercial. Exemplo deste interesse foi o "*Netflix prize*" promovido pela Netflix, empresa de distribuição *online* de filmes, que em 2006 anunciou prêmio de um milhão de dólares a quem desenvolvesse um algoritmo de recomendação de filmes que fosse 10% mais preciso que o algoritmo Cinematch, usado na ocasião pela empresa. O prêmio foi recebido em 2009 pela equipe BellKor's Pragmatic Chaos, criada ao longo do concurso pela fusão de diferentes grupos competidores que reuniram esforços para criar o *ensemble* de algoritmos vencedor [8].

Os sistemas de recomendação baseados em filtragem colaborativa são abertos por natureza, já que as se baseiam nas avaliações de uma comunidade de usuários, o que os torna vulneráveis a ataques de injeção de perfis. Esse tipo de ataque consiste em inserir perfis tendenciosos na base de dados do sistema permitindo que atacantes possam manipular as recomendações. O objetivo desses ataques pode ser tanto promover itens (*push*) quando ofuscá-los (*nuke*), tendo como alvo um usuário ou um grupo de usuários [9].

Nesse contexto, analisar a robustez e a estabilidade de algoritmos de recomendação passa a ser uma questão relevante. Ao analisar a robustez, mede-se o desempenho do sistema antes e depois do ataque, para verificar como o ataque afetou o sistema como um todo. Na análise da estabilidade deseja-se verificar o desvio dos valores previstos pelo sistema para os itens atacados.

Em trabalhos anteriores foram propostos algoritmos para a recomendação de serviços Web com marcação semântica [10][11]. As avaliações realizadas, sem considerar a possibilidade de ataques, mostraram que os algoritmos têm bom desempenho se avaliados sob a perspectiva da precisão das recomendações feitas, especialmente em situações em que a matriz usuário-item, que armazena as avaliações feitas pela comunidade de usuários, é esparsa. Diante disso, o objetivo da pesquisa relatada neste artigo é analisar a estabilidade e robustez desses algoritmos quando sujeitos a ataques de injeção de perfis.

As demais seções do artigo estão organizadas da seguinte maneira: na seção II são apresentados os algoritmos de filtragem colaborativa de serviços Web semânticos cuja estabilidade e robustez se deseja estudar; na seção III serão melhor caracterizados os ataques de injeção de perfis objeto deste trabalho; na seção IV apresentam-se e discutem-se o método de trabalho empregado para verificar a robustez e estabilidade dos algoritmos considerados e os experimentos conduzidos com esse propósito; na seção V apresentam-se alguns trabalhos correlatos; e, na seção VI, encerra-se o artigo retomando as principais conclusões e fazendo algumas considerações finais.

II. Filtragem Colaborativa de Serviços Web Semânticos

A. *Serviços Web Semânticos*

Os algoritmos tratados neste artigo visam à recomendação de serviços Web marcados semanticamente segundo a ontologia de anotação de serviços OWL-S, definida a partir da linguagem de descrição de ontologias OWL, padrão do W3C, baseada em lógica de primeira ordem [12]. A ontologia OWL-S consiste de três partes principais: o perfil do serviço, usado para anunciar e descobrir serviços; o modelo do processo, que fornece uma descrição detalhada do funcionamento do serviço; e um aterramento (*grounding*), que fornece detalhes de como interoperar com um serviço através de mensagens. A maioria dos sistemas de casamento de serviços descritos em OWL-S, empregados para comparar e buscar serviços Web semânticos utiliza apenas o perfil do serviço. Este perfil especifica a semântica a assinatura do serviço, isto é das entradas requeridas pelo serviço e das saídas produzidas. O perfil também permite descrever as precondições a serem satisfeitas para que o serviço possa ser executado e descrever os resultados esperados da execução do serviço. Estas informações são geralmente conhecidas pela sigla IOPE (*inputs*, *outputs*, *preconditions*, *effects*). Com isso, podem-se utilizar métodos de raciocínio lógico para determinar o grau de similaridade entre dois serviços.

Os algoritmos usados neste trabalho para verificar o grau de similaridade entre dois serviços utilizam apenas a assinatura dos serviços, isto a descrição das entradas e das saídas. Isto certamente permite que ocorram falsos positivos em alguns casamentos, mas isso não se mostrou um problema relevante. Os parâmetros de entrada e saída são associados a conceitos de ontologias de domínio, tais como Pessoa, Médico, Veículo, Veículo-Motorizado, e não como tipos básicos (int, char, real) ou meras seqüências de caracteres. Com isso não se pode garantir que se esteja comparando serviços que fazem coisas diferentes com entradas que representam o mesmo conceito, mas reduzem bastante essa possibilidade. Naturalmente que o que se deseja é ter algoritmos de casamento que não se limitem à comparação das entradas e saídas. Isso, no entanto, não é uma preocupação desta pesquisa, e não deve alterar substancialmente os resultados verificados experimentalmente.

A OWL-S, assim como a OWL, é baseada na lógica de descrição. Sabe-se que as lógicas formais têm capacidades limitadas de expressão. A lógica de descrição, por exemplo, não permite representar com precisão objetos estruturados interconectados arbitrariamente. De modo a contornar parcialmente essa capacidade limitada de representação do OWL-S, neste trabalho utilizou-se o mecanismo híbrido de casamento se serviços Web semânticos implementado no sistema OWLS-MX [13]. Esse mecanismo permite comparar dois serviços Web semânticos e estabelecer quatro graus de

similaridade semântica usando raciocínio lógico, e um valor de similaridade sintática usando métodos de recuperação de informação.

B. Filtragem Colaborativa

Algoritmos de filtragem colaborativa podem ser usados para prever o valor que um determinado usuário (usuário alvo) atribuiria a um item (item alvo) ainda não avaliado por esse usuário. A previsão é feita utilizando o histórico de avaliações feitas pela comunidade de usuários para os itens considerados.

Podem ser usados algoritmos muito simples como IMEAN, que estima os valores de avaliações desconhecidas como sendo a média dos valores das avaliações feitas por todos os usuários para o item alvo, ou o UMEAN, que estima a avaliação de um item alvo a partir da média de todos os itens avaliados pelo usuário alvo. Esses algoritmos são de fácil implementação e baixo custo de execução, mas pouco precisos, servindo, em geral, apenas com termo de comparação para a avaliação de algoritmos mais elaborados, como os outros dois algoritmos avaliados neste trabalho: o primeiro baseado no algoritmo dos *k* vizinhos mais próximos (*k-nn*) e o segundo no método de agrupamento *k-médias (k-means).*

C. Previsão de Avaliações Usando o K-nn

O algoritmo *k-nn* é um algoritmo baseado em memória, e, como tal, prevê a nota que um usuário daria a um item usando diretamente as avaliações dos *k* usuários mais semelhantes a esse usuário.

Na implementação considerada, a similaridade entre dois usuários *u* e *v*, $s_{u,v}$ é computada usando o coeficiente de correlação de Pearson (CCP), estendido para considerar serviços similares, conforme a equação (1).

$$s_{u,v} = \frac{\sum_{s \in S_u, t \in S_v} (f_{u,s} - \overline{f}_u)(f_{v,t} - \overline{f}_v)}{\sqrt{\sum_{s \in S_u} (f_{u,s} - \overline{f}_u)^2} \sqrt{\sum_{t \in S_v} (f_{v,t} - \overline{f}_v)^2}} \qquad (1)$$

Na equação (1), S_u é o conjunto de serviços avaliados por *u* e S_v é o conjunto de serviços acessados por *v*; $\overline{f}_u$ e $\overline{f}_v$ são os valores médios das avaliações feitas pelos usuários *u* e *v*, respectivamente; entre os serviços avaliados por *v*, *t* é o serviço que apresenta maior similaridade semântica ou sintática com *s* (serviço acessado e avaliado por *u*), respeitado um grau de similaridade mínimo. Quando ambos usuários avaliaram um mesmo serviço, *s e t* representam o mesmo serviço.

Cabe observar que o CCP não dá pesos diferentes a usuários semelhantes que avaliaram um conjunto reduzido de itens de forma semelhante ou que avaliaram um conjunto muito grande. Igualmente, a equação (1) não diferencia entre usuários cuja similaridade se deve a avaliações dos mesmos itens e às avaliações de itens similares. Estas são questões não foram tratadas neste trabalho, mas que são interessantes o bastante para merecer atenção em pesquisas futuras.

A Tabela I apresenta um exemplo de matriz usuário-item, para itens genéricos, não necessariamente serviços, sem mostrar informação que permita verificar a semelhança semântica ou sintática entre os itens. Na tabela, as avaliações variam de 1 a 5. Uma posição vazia indica que o usuário correspondente àquela linha não avaliou o item correspondente à coluna. A matriz registra os valores das avaliações feitas por seis usuários para seis itens quaisquer (livros, filmes, serviços Web etc.). A coluna à direita da matriz mostra a similaridade entre o usuário 1 e os demais usuários, utilizando a correlação de Pearson. Se for considerado apenas o usuário mais próximo ao usuário 1 (k=1), um algoritmo baseado no método *k-nn* preveria que usuário 1 atribuiria ao item alvo (item 6) um valor próximo ao atribuído a esse item pelo usuário 6 (o mais similar ao usuário 1).

TABELA I
EXEMPLO DE UMA MATRIZ USUÁRIO-ITEM

	Itens						***Similaridade com o usuário 1***
Usuários	***1***	***2***	***3***	***4***	***5***	***6***	
1	*5*	*2*	*3*	*3*		*?*	*1,00*
2	*2*		*4*		*4*	*1*	*- 1,00*
3	*3*	*1*	*3*		*1*	*2*	*0,76*
4	*4*	*2*	*3*	*1*		*1*	*0,72*
5	*3*	*3*	*2*	*1*	*3*	*1*	*0,21*
6	*4*	*3*		*3*	*3*	*2*	*0,94*

Conhecendo a similaridade entre usuários, pode-se estimar a avaliação que um usuário faria de um determinado serviço alvo se usuários similares tiverem avaliado esse serviço ou serviços semelhantes. Para isso, define-se a vizinhança *V* do usuário *u* com respeito ao serviço *s* como sendo formada pelos *k* usuários mais similares a *u* que acessaram o serviço *s* ou serviços similares a *s*. Construída a vizinhança, a previsão da avaliação que o usuário u faria de um serviço *s*, $f_{u,s}$, é feita usado a média ponderada de todas as avaliações do serviço *s*, ou de serviços similares, feitas pelos usuários em *V*, conforme a equação (2)

$$f_{u,s} = \overline{f}_u + \frac{\sum_{v \in V} s_{u,v} (f_{v,t} - \overline{f}_v)}{\sum_{v \in V} |s_{u,v}|} \qquad (2)$$

Na equação (2), assim como na equação (1), *t* é o serviço acessado por *v* mais similar ao serviço *s* (acessado por *u*), respeitado um patamar de similaridade mínima. Se *V* for vazio, $f_{u,s}$ é estimado como sendo igual a $\overline{f}_u$ (a média de todas as avaliações feitas pelo usuário *u).*

Aplicando a equação (2) aos dados da Tabela I, e considerando *k*=1, o valor previsto para a avaliação que o usuário 1 faria do item 6 seria:

$f_{1,6}$ = 13/4 + (0,94 * (2 - 15/5))/ 0,94 = 2,25

D. Previsão de Avaliações com o Algoritmo K-médias

A literatura aponta que algoritmos de filtragem colaborativa baseados em memória, como o *k-nn*, costumam apresentar

precisão elevada, mas baixa escalabilidade, por fazerem as previsões diretamente a partir dos dados disponíveis. Entretanto, isto demanda um tempo de processamento elevado a cada previsão para identificar a vizinhança do usuário alvo para cada item (serviço) alvo considerado. Com alternativa, nos algoritmos baseados em modelo, as previsões não são feitas diretamente a partir das avaliações dos usuários mais próximos ao usuário alvo, mas sim de um modelo construído previamente a partir dos dados disponíveis.

Analisa-se neste artigo um algoritmo de filtragem colaborativa baseado em modelo construído a partir do método de agrupamento *k-médias*. O algoritmo agrupa perfis de usuários semelhantes em grupos e faz as previsões usando os perfis dos grupos, representados pelos seus centróides. Neste contexto, o perfil de um usuário é definido pelas avaliações que o usuário realizou dos itens disponíveis (uma linha da matriz usuário-item). Por exemplo, na matriz usuário-item da Tabela I, o perfil do usuário 1 pode ser representado pela n-upla (5,2,3,3, Ø, Ø), onde Ø indica que o item correspondente não foi avaliado.

O funcionamento do algoritmo é o seguinte: inicialmente *k* pontos (perfis de usuários definidos pelas n-uplas que contêm as avaliações atribuídos pelos usuários aos itens) são escolhidos como centróides de *k* grupos, onde *k* é um parâmetro previamente definido. A seguir, um passo de atribuição e um passo de atualização são repetidos até a convergência do algoritmo. No passo de atribuição, cada ponto (perfil) é associado ao grupo com o centróide mais próximo. No passo de atualização, os centróides dos grupos são atualizados para a média dos pontos associados ao grupo. O algoritmo converge quando os centróides tornam-se estáveis, ou seja, não mudam na fase de atualização. Na implementação sob estudo, a equação (1) é usada para calcular a distância entre um usuário e o centróide do grupo. Definidos os grupos, a equação (2) é então usada para prever a avaliação que um usuário faria de um item, usando uma vizinhança constituída pelos centróides dos grupos mais próximos ao usuário alvo, em lugar das avaliações (n-uplas) dos usuários mais próximos. Como o número de centróides de grupos contido na vizinhança considerada é, geralmente, muito menor que o número de usuários que formam uma vizinhança no algoritmo baseado no *k-nn*, o algoritmo baseado no *k-médias* tem um tempo de computação sensivelmente inferior, conforme comprovado em experimentos não relatados neste artigo.

III. ATAQUE DE INJEÇÃO DE PERFIS

Em um ataque de injeção de perfis, o atacante insere perfis tendenciosos na matriz usuário-item do sistema de recomendação.

A Tabela II mostra a matriz usuário-item da Tabela I na qual foi inserido um perfil falso, o usuário 7. Feita essa inserção, o usuário mais próximo ao usuário 1 passa a ser o atacante, com grau de similaridade entre eles igual a 0,98.

Nestas condições, se for usado um algoritmo do tipo *k-nn*, com *k*=1, o valor da avaliação estimada a ser atribuída ao item 6 pelo usuário 1 seria igual a 4,85. Isso pode ser verificado se considerarmos que, usando a equação (2), temos:

- $f_{u,s} = f_{1,6}$
- $\overline{f_u} = (5+2+3+3)/4 = 13/4$; média de todas as avaliações feitas pelo usuário *u=1*
- $s_{u,v} = 0{,}98$; grau de similaridade entre os usuários u=1 e v=7, conforme a tabela.
- $f_{v,t} = 5$; *v=7 e t = 6, pois* o usuário 7 acessou o serviço 6
- $\overline{f_v} = (4+2+3+3+5)/5 = 17/5$; média das avaliações feitas pelo usuário v=7

Logo, $f1{,}6 = 13/4 + (0{,}98 * (5 - 17/5)) / 0{,}98 = 4{,}85$

Embora não seja prática corrente usar uma vizinhança tão reduzida, o exemplo acima visa a enfatizar o impacto negativo que um ataque bem sucedido pode ter.

TABELA II
EXEMPLO DE UMA MATRIZ USUÁRIO-ITEM COM UM PERFIL DE ATAQUE (USUÁRIO 7)

Usuários	Itens						Similaridade com o usuário 1
	1	2	3	4	5	6	
1	5	2	3	3		?	1,00
2	2		4		4	1	- 1,00
3	3	1	3		1	2	0,76
4	4	2	3	1		1	0,72
5	3	3	2	1	3	1	0,21
6	4	3		3	3	2	0,94
7	**4**	**2**		**3**	**3**	**5**	**0,98**

A. *Tipos de Ataque de Injeção de Perfis*

Ataques de injeção de perfis podem ser caracterizado por quatro conjuntos de itens:

- um conjunto unitário contendo o item alvo i_t;
- um conjunto de itens selecionados com características determinadas pelo atacante, I_S;
- um conjunto de itens de preenchimento, I_F, geralmente escolhidos de forma aleatória, e
- um conjunto de itens não avaliados, $I_Ø$.

Tipos de ataques de injeção de perfis são definidos pelos métodos usados para identificar os itens selecionados, a proporção de itens de preenchimento e pelo modo de determinar as avaliações associadas a cada um desses conjuntos de itens e ao item alvo.

Neste artigo, serão realizados experimentos para procurar reproduzir dois dos mais representativos tipos de ataques de injeção de perfis: o ataque aleatório (*random attack*) e o ataque a segmento (*segment attack*).

No ataque aleatório, I_S é vazio, os itens em I_F são preenchidos aleatoriamente segundo uma distribuição normal em torno da avaliação média de todos os itens na base de dados, e i_t é preenchido com o valor máximo que pode ser atribuído a uma avaliação. O conhecimento requerido para

montar um ataque aleatório é muito reduzido, já que a média geral das avaliações dos itens de um sistema de recomendação geralmente pode ser determinada empiricamente sem grande dificuldade por um observador externo, e, em muitos casos, está disponível diretamente através do sistema. Entretanto, o custo de execução deste ataque pode ser elevando, pois é necessário calcular avaliações específicas (aleatórias) para cada item dos perfis de ataque.

No ataque a segmento (*segment attack*), os itens em I_S definem uma categoria, ou segmento. Esses itens recebem avaliações máximas quando se deseja promover um item (*push*), ou mínimas quando se deseja obscurecê-lo (*nuke*). Se esses itens forem, por exemplo, serviços de viagens, os itens em I_S corresponderiam a serviços dessa categoria. Esse ataque é notavelmente efetivo se o seu alvo pertencer ao segmento considerado. Os valores em I_F são preenchidos com a avaliação mínima e i_t com a avaliação máxima. A literatura destaca que este é um ataque bastante eficaz que requer pouco conhecimento para ser engendrado [14].

IV. AVALIAÇÃO DA ESTABILIDADE E ROBUSTEZ DOS ALGORITMOS DE RECOMENDAÇÃO

Conforme enunciado na seção introdutória, o objetivo da pesquisa relatada neste artigo é analisar a estabilidade e robustez de algoritmos para a recomendação de serviços Web com marcação semântica, quando sujeitos a ataques de injeção de perfis. Para atingir esse objetivo foi realizada uma pesquisa experimental que compreendeu os seguintes passos:

i. seleção de base de serviços com marcação semântica;
ii. criação de conjunto de dados para treinamento e teste dos algoritmos, contendo perfis de usuários caracterizados pelas avaliações por eles feitas a uma seleção dos serviços da base identificada na etapa anterior;
iii. seleção de usuários e itens a serem vítimas do ataque;
iv. aplicação dos algoritmos de recomendação, sem a presença de perfis de ataque, e medição do erro médio absoluto normalizado (NMAE) das previsões feitas pelos algoritmos;
v. injeção de perfis atacantes;
vi. nova aplicação dos algoritmos de recomendação e determinação do NMAE e de outras métricas que medem o sucesso dos ataques;
vii. análise dos resultados obtidos.

Esses passos e seus resultados são apresentados e discutidos com mais detalhes nesta seção. Inicialmente, a subseção A, apresentará as métricas utilizadas para medir o sucesso dos ataques. Em seguida, a subseção B, descreverá os experimentos conduzidos, e, a subseção C, apresentará e analisará os resultados obtidos nos experimentos.

A. *MÉTRICAS DE AVALIAÇÃO*

A eficácia de ataques de injeção de perfis sobre itens específicos pode ser avaliada examinando o desvio de previsão induzido pelo ataque, a taxa de acerto do ataque e a influência do ataque sobre o erro absoluto médio das previsões realizadas.

Dado um usuário u e um item i, o desvio de previsão, $\Delta_{u,i}$, é calculado pela equação (3). Nessa equação, $P'(r_{u,i})$ é o valor previsto para a avaliação que o usuário u faria do item i depois do ataque e $P(r_{u,i})$ representa o valor previsto antes do ataque.

$$\Delta_{u,i} = P'(r_{u,i}) - P(r_{u,i}) \quad (3)$$

Um desvio de previsão positivo indica que o ataque teve sucesso em fazer o item melhor avaliado. Todavia, um aumento elevado do desvio de previsão para um item não garante que o item será recomendado. É possível que outros itens também sejam afetados pelo ataque ou que o item fosse inicialmente avaliado com um valor muito baixo, de modo que mesmo um desvio elevado não o coloque na categoria de recomendável.

Para avaliar que itens atacados foram efetivamente recomendados pode-se usar a taxa de acerto (Hit Ratio- HR), que mede a eficácia do ataque sobre um item. A taxa de acerto para o item i é calculada pela equação (4), onde $H_{u,i}$ será igual a 1 se o item i aparecer na lista dos N itens recomendados ao usuário u (*top-N recommendations*), em caso contrário seu valor será 0; U_T é o conjunto de usuários alvo.

$$HR_i = \sum_{u \in U_T} H_{u,i} \Big/ U_T \quad (4)$$

O efeito do ataque considerando todas as avaliações previstas pode ser avaliado pelo erro médio absoluto (*Mean Absolute Error – MAE*) ou pelo erro médio absoluto normalizado (*Normalized Mean Absolute Error – NMAE*), métricas geralmente usadas para avaliar a precisão das previsões feitas por algoritmos de recomendação. O *MAE* é obtido a partir da diferença entre os valores previstos e os valores reais das avaliações, conforme mostrado na equação (5), onde $r_{u,i}$ é a nota real atribuída pelo usuário u ao item i, $r'_{u,i}$ a nota prevista pelo algoritmo para esse item e N o número de notas previstas pelo algoritmo.

$$MAE = \frac{\sum_{u,i} \left| r_{u,i} - r'_{u,i} \right|}{N} \quad (5)$$

O *MAE* pode ser normalizado de modo a torná-lo independente da escala de avaliações usada, dando origem ao erro médio absoluto normalizado, ou *NMAE* (*Normalized Mean Absolute Error*), calculado pela equação (6).

$$NMAE = \frac{MAE}{\sum_{u,i} r_{u,i} \Big/ N} \quad (6)$$

B. CARACTERIZAÇÃO DOS EXPERIMENTOS

Os experimentos realizados tiveram por objetivo avaliar o impacto dos ataques de injeção de perfis dos tipos aleatório e segmento sobre os algoritmos de filtragem colaborativa IMEAN, UMEAN, *k-nn* e *k-médias* apresentados na seção II. O desempenho desses algoritmos sem a presença de ataques, em especial quando a matriz usuário-item é esparsa, é analisado em [11].

Uma das principais dificuldades para a avaliação de algoritmos para a recomendação de serviços Web semânticos é necessidade de uma base pública de serviços com marcação semântica. O mais próximo que se tem disso, e que se usou neste trabalho, é a coleção OWLS-TC[1], criada para avaliar o desempenho de algoritmos de casamento (*matchmaking*) de serviços Web semânticos descritos segundo a ontologia de anotação de serviços OWL-S 1.1. Nos experimentos descritos, foi utilizada a versão 2.2 dessa base que reúne 1004 serviços Web de vários domínios.

Para realizar os experimentos foram criados 50 usuários, 20 com o perfil de "turista" e 30 com o perfil de "estudante". Os itens a serem avaliados totalizam 108 serviços, incluindo serviços relacionados aos assuntos "carros", "comida", "livros", "hotéis", "câmeras", "publicações", "surf" e "filmes". Cada usuário avaliou 18 serviços que seriam característicos do seu perfil. Um turista, por exemplo, se interessaria mais em hotéis que estudantes, e, portanto, há a possibilidade de encontrar mais avaliações de hotéis no grupo dos turistas que no grupo dos estudantes.

Na montagem dos ataques foram usados parâmetros semelhantes aos empregados em [14]. Desse modo, foi inserida na matriz usuário-item uma quantidade de perfis de ataque igual a 15% do total de usuários, resultando na injeção de oito usuários atacantes, todos com o objetivo de promover um item (*push*), o objetivo mais comum de ataques. Para alvos, foram escolhidos aleatoriamente cinco usuários e quatro serviços, que correspondem a aproximadamente 5% do total de usuários e 3% do total de itens. Nos perfis de ataque, os itens de preenchimento correspondem a 6% do total de serviços (I_F). A escolha dos itens para preenchimento no ataque aleatório é feita automaticamente, após especificar a quantidade de perfis a serem criados. No ataque a segmento, foi definido um segmento caracterizado por três itens (I_S). O segmento escolhido para os ataques foi o de "comida", pois se presume que tanto estudantes quanto turistas se interessem por serviços relacionados a esse assunto, de modo que essa escolha deve afetar a ambos os grupos de usuários.

C. RESULTADOS E DISCUSSÕES

Os experimentos cujos resultados são apresentados nesta seção foram repetidos 10 vezes, de modo que os valores apresentados para o *NMAE*, taxa de acerto e desvio de previsão constituem a média dos valores obtidos nessas repetições.

Influência dos ataques no NMAE

A *Tabela III* apresenta o *NMAE* para os algoritmos estudados antes e depois dos ataques dos tipos aleatório e segmento.

Conforme pode ser observado na *Tabela III*, o UMEAN não é afetado pelos ataques, o que faz sentido, já que a adição de perfis não interfere com esse algoritmo, pois a avaliação prevista para um item é a média das avaliações feitas pelo usuário alvo, não sendo consideradas as avaliações dos usuários atacantes.

Para o algoritmo IMEAN, esperava-se que os ataques tivessem grande impacto no *NMAE* e na taxa de acerto, já que suas previsões são baseadas nas notas de toda a comunidade de usuários. A *Tabela III* e a Fig. 1 mostram que essa expectativa se concretizou. Na *Tabela III* pode-se observar que para o IMEAN o ataque aleatório aumentou o NMAE em 45,9% (de 0,33064 para 0,48232) e o ataque a segmento em 44,7% (de 0,33064 para 0,47835).

TABELA III
NMAE DOS ALGORITMOS AVALIADOS

	Sem ataque	*Ataque aleatório*	*Ataque a segmento*
UMEAN	*0,33111*	*0,33111*	*0,33111*
IMEAN	*0,33064*	*0,48232*	*0,47835*
k-nn	*0,27781*	*0,28239*	*0,28415*
k-médias	*0,29305*	*0,30015*	*0,30193*

No caso dos algoritmos *k-nn* e *k-médias*, os ataques resultaram em aumentos perceptíveis no *NMAE*, embora muito menos expressivos do que verificado para o IMEAN. O ataque aleatório aumentou o *NMAE* em 1,6% para o *k-nn* (de 0,27781 para 0,28239) e 2,4% para o *k-médias* (de 0,29305 para 0,30015); enquanto o ataque a segmento levou a aumentos de 2,3% para o *k-nn* (de 0,27781 para 0,28415) e 3,0% para o *k-médias* (de 0,29305 para 0,30193).

Taxa de Acerto e Desvio de Previsão

A Fig. 1 apresenta a taxa de acerto dos ataques para o algoritmo IMEAN. O eixo das abscissas do gráfico indica quantos itens serão recomendados para o usuário, os *top-N (*os N itens com as maiores avaliações previstas), e o eixo das ordenadas representa o taxa de acerto. Quanto mais próxima de um for a taxa de acerto, maior o percentual de itens atacados que terá sido recomendado aos usuários alvo. Uma taxa de acerto igual a um significa que todos os itens alvo do ataque foram recomendados.

[1] http://projects.semwebcentral.org/frs/?group_id=89

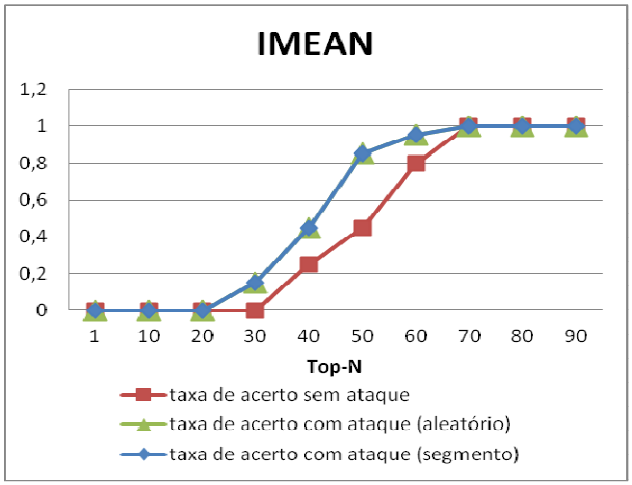

Fig. 1. Taxa de acerto ao usar o algoritmo IMEAN

Observa-se na Fig. 1 que tanto o ataque aleatório quanto o ataque a segmento apresentaram resultados semelhantes para o IMEAN. Isso acontece, pois esse algoritmo usa as avaliações de todos os usuários para o item alvo, o que leva os dois tipos de ataques a efeitos semelhantes. Para *top-n* de até 20 itens e superior a 70 os efeitos dos ataques foram pouco perceptíveis, porém para *top-n* entre 20 e 70 a taxa de acerto foi elevada, o que implica em índices de sucesso elevados na inclusão dos itens atacados nas listas de recomendação dos usuários.

Ambos ataques produziram em um desvio de previsão igual a 2,70966 (não apresentado nas figuras), bastante elevado se comparados com os valores observados para os algoritmos *k-nn* e *k-médias*, apresentados a seguir.

Para medir o desvio de previsão dos algoritmos *k-nn* e *k-médias*, variou-se o valor de semelhança mínima entre os usuários (*k-nn*) e entre um usuário e o centróide de um grupo (*k-médias*) usada no momento de escolher os k vizinhos mais próximos e, assim, aplicar a equação (2). Além disso, deixou-se fixo em dois o número de *clusters* utilizado durante a

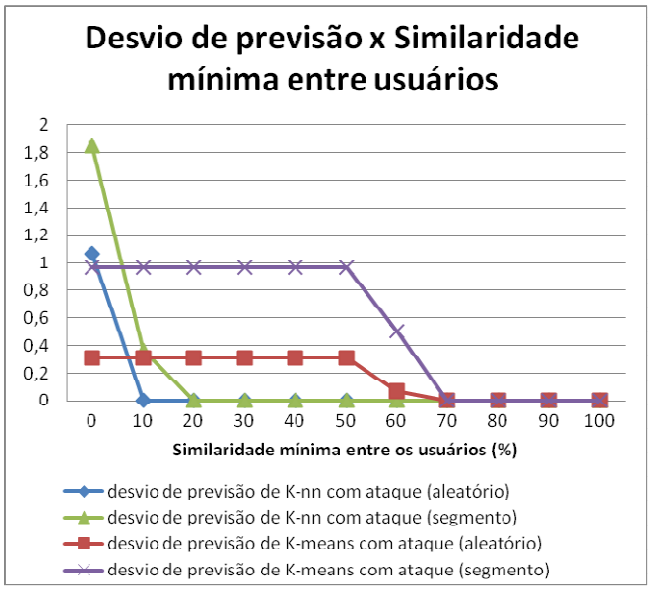

Fig. 2. Desvio de previsão dos algoritmos *k-nn* e k-médias em função da similaridade mínima entre usuários

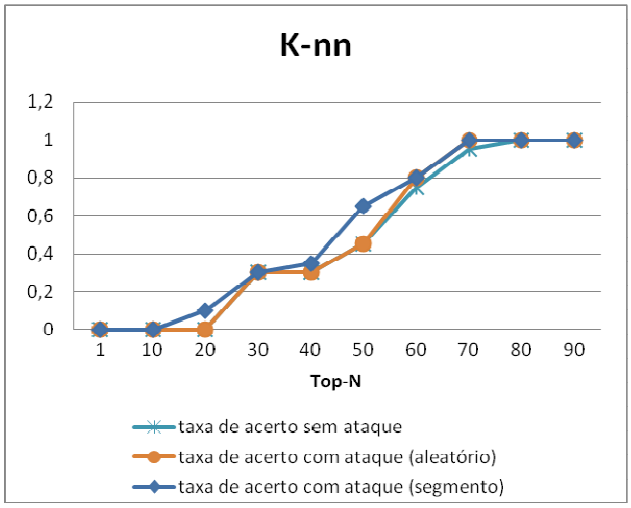

Fig. 3. Taxa de acerto do algoritmo *k-nn*

realização dos testes com o *k-médias* (o valor de k foi escolhido após uma série preliminar de experimentos de ajuste). A Fig. 2 apresenta os resultados obtidos. Pode-se observar que os desvios de previsão para os algoritmos *k-nn* e *k-médias* são sensivelmente menores que os verificados para o UMEAN e IMEAN. Nota-se também que a partir de uma semelhança entre usuários de 20% para o algoritmo *k-nn* e de 70% para o *k-médias*, o desvio de previsão permanece com valor zero, o que significa que, a partir desses valores, a semelhança exigida foi suficiente para que não fossem incluídos os perfis de ataque no cálculo de previsão das avaliações dos itens.

Na análise da taxa de acerto dos ataques quando se usa o *k-nn* e o *k-médias* foram usados valores de similaridade que resultaram em um desvio de previsão alto e que estivessem próximos ao valor em que o desvio cai a zero. Os valores adotados foram de 5% para o algoritmo *k-nn* e 50% para o algoritmo *k-médias*.

Nota-se na Fig. 3, que mostra a taxa de acerto para o *k-nn*, que os ataques aleatório e a segmento tiveram algum sucesso

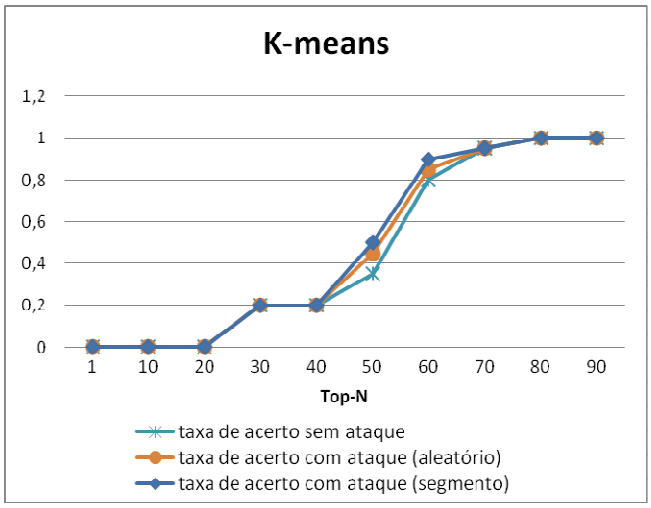

Fig. 4. Taxa de acerto do algoritmo k-médias

para os *top* 60 e 70, mas o ataque a segmento teve uma taxa de acerto elevada para os top 20 e 50.

A Fig. 4 mostra que para o *k-médias* só se observa uma taxa de acerto apreciável nos *top* 50 e 60, novamente com destaque para o ataque a segmento.

V. TRABALHOS RELACIONADOS

O ataque aleatório foi originalmente proposto por Lam and J. Riedl [25]. O conhecimento requerido para montar esse ataque é muito reduzido, mas o custo de execução pode ser alto, já que é necessário atribuir avaliações a cada item no perfil de ataque. Por outro lado, os autores mostraram, e nossos resultados conformaram, que o ataque não é muito eficaz.

O ataque a segmento foi introduzido por Mobasher et al. em [14]. Os autores mostraram, e os nossos experimentos confirmam, que é possível executar ataques desse tipo bem sucedidos contra sistemas de recomendação baseados em filtragem colaborativa sem a necessidade de ter um conhecimento substancial sobre o sistema ou sobre os usuários.

Vários trabalhos, incluindo [14][17][20][21], mostraram que ataques de injeção de perfis podem afetar significativamente a robustez de sistemas de recomendação. Isso tem levado vários autores à busca por algoritmos de recomendação mais robustos e estáveis utilizando diversos mecanismos, incluindo limitadores da influência dos atacantes [18], seqüências dinâmicas de avaliações em lugar de conjuntos estáticos de perfis de avaliações [22] e oferta de incentivos monetários para que outros avaliadores corrijam as distorções do sistema, provocadas ou não por ataques [23]. Diversos autores propõem ainda a utilização de estratégias para a detecção de ataques, utilizando, entre outros, mecanismos de aprendizado não supervisionado e semi-supervisionado [16][24] e modelos estatísticos [19].

Neste artigo o termo estabilidade está associado a medir a dinâmica das previsões do sistema quando sofre ataques externos. Ao medir a estabilidade, avalia-se a mudança ocorrida nas previsões do sistema para os itens atacados. Alguns autores, como Adomavicius e Zhang [15], observam um aspecto diferente da estabilidade, denominada de consistência interna, que representa as conseqüências das inconsistências internas dos algoritmos do sistema de recomendação. Para esses autores, um algoritmo de recomendação estável oferece previsões consistentes ao longo do tempo, assumindo que as novas avaliações que se tornam disponíveis estão de acordo com as previsões prévias do sistema.

VI. CONCLUSÃO

Os experimentos realizados para avaliar a estabilidade e robustez dos algoritmos IMEAN, UMEAN, *k-nn* e *k-médias* para a recomendação de serviços Web semânticos, quando submetidos a ataques de injeção de perfis, mostraram que o algoritmo UMEAN não é afetado pelos ataques e que o IMEAN é o mais sensível a esses ataques. Entretanto esses dois algoritmos são usados apenas para fornecer limites inferiores de desempenho para a análise dos demais algoritmos, já que ambos apresentam baixa precisão, particularmente quando a matriz usuário-item é esparsa, como mostrado em [10] e [11]. Para os demais algoritmos, verificou-se que o *k-nn* permite atingir um desvio de previsão próximo a zero para taxas de semelhança entre usuários consideravelmente reduzida (20%), enquanto que no *k-médias* isso só é possível para valores mais elevados (70%). Por outro lado, nos experimentos apresentados para analisar a taxa de acerto, o *k-médias* apresenta melhores resultados em comparação com o *k-nn*, quando o *k-nn* usa um valor de semelhança entre usuários muito inferior ao utilizado no *k-médias*. Com valores elevados de semelhança entre usuários, o *k-nn* apresenta melhores resultados também neste quesito. Apesar destes resultados, na escolha de qual algoritmo deve ser empregado em uma aplicação concreta deve-se considerar também que, segundo a literatura, o *k-médias* costuma ser mais escalável que o *k-nn*. Com relação à eficácia dos ataques, os experimentos mostram que o ataque a segmento é mais efetivo que o aleatório.

Os algoritmos foram também avaliados considerando a utilização da similaridade semântica e sintática entre serviços Web quando do cálculo da semelhança entre usuários utilizando a equação (1) e no momento de prever o valor de uma avaliação com a equação (2). Os experimentos mostraram que isso não influenciou de forma significativa os resultados, o que é coerente com os resultados encontrados em [10], onde se verificou que a utilização da similaridade semântica entre serviços afeta de forma relevante a precisão dos algoritmos apenas quando a matriz usuário-item é esparsa, o que não se verifica nos experimentos realizados, onde esta matriz era densa.

Em trabalhos futuros deverão serão feitos experimentos para avaliar o comportamento dos algoritmos quando submetidos a ataques considerando conjuntos de dados de maiores dimensões e matrizes usuário-item com dados esparsos.

AGRADECIMENTOS

Agradecemos aos revisores anônimos e ao editor da revista, Prof. Dr. Ricardo Linden, pela cuidadosa revisão e valiosas correções, recomendações e comentários que muito contribuíram para aumentar a clareza e precisão do artigo.

REFERÊNCIAS

[1] H. Chesbrough and J. Spohrer, “A research manifesto for services science,” Commun. ACM, vol. 49, no. 7, pp. 35–40, 2006.

[2] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, “Web Services Description Language (WSDL) 1.1, 2001,” At http://www. w3. org/TR/2001/NOTE-wsdl-20010315, 2001.

[3] L. Clement, A. Hately, C. von Riegen, and T. Rogers, “UDDI Version 3.0.” OASIS, 19-Oct-2004.

[4] S. A. McIlraith, T. C. Son, and H. Zeng, “Semantic web services,” Intelligent Systems, IEEE, vol. 16, no. 2, pp. 46–53, 2005.

[5] T. Berners-Lee, J. Hendler, and O. Lassila, “The semantic Web,” Scientific American, vol. 284, no. 5, pp. 28–37, 2001.

[6] R. M. Sreenath and M. P. Singh, “Agent-based service selection,” Web Semantics: Science, Services and Agents on the World Wide Web, vol. 1, no. 3, pp. 261–279, 2004.
[7] X. Su and T. M. Khoshgoftaar, “A survey of collaborative filtering techniques,” Advances in Artificial Intelligence, vol. 2009, pp. 1–19, 2009.
[8] S. Lohr, “The contest that shaped careers and inspired research papers,” CHANCE, vol. 23, no. 1, pp. 25–29, 2010.
[9] J. J. Sandvig, B. Mobasher, and R. Burke, “A survey of collaborative recommendation and the robust-ness of model-based algorithms,” IEEE Data Engineering Bulletin, vol. 31, no. 2, pp. 3–13, 2008.
[10] J. M. Adán-Coello, Y. Yuming and C. M. Tobar, “A Memory-based Collaborative Filtering Algorithm for Recommending Semantic Web Services. Revista IEEE América Latina, v. 11, p. 795-801, 2013.
[11] J. M. Adán-Coello, C. M. Tobar and Y. Yuming , “Improving the Performance of Web Service Recommenders Using Semantic Similarity”. Submetido.
[12] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, S. McIlraith, S. Narayanan, M. Paolucci, B. Parsia, T. Payne, and others, “OWL-S: Semantic Markup for Web Services,” 2004.
[13] M. Klusch, B. Fries, and K. Sycara, “OWLS-MX: A hybrid Semantic Web service matchmaker for OWL-S services,” Web Semantics: Science, Services and Agents on the World Wide Web, vol. 7, no. 2, pp. 121–133, Apr. 2009.
[14] B. Mobasher, R. Burke, R. Bhaumik, and C. Williams, “Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness,” ACM Transactions on Internet Technology (TOIT), vol. 7, no. 4, p. 23, 2007.
[15] G. Adomavicius and J. Zhang, “Stability of recommendation algorithms,” ACM Transactions on Information Systems (TOIS), vol. 30, no. 4, p. 23, 2012.
[16] R. Bhaumik, B. Mobasher, and R. D. Burke, “A clustering approach to unsupervised attack detection in collaborative recommender systems,” in Proceedings of the 7th IEEE international conference on data mining, Las Vegas, NV, USA, 2011, pp. 181–187.
[17] G. Shani and A. Gunawardana, “Evaluating recommendation systems,” Recommender Systems Handbook, pp. 257–297, 2011.
[18] P. Resnick and R. Sami, “The influence limiter: provably manipulation-resistant recommender systems,” in Proceedings of the 2007 ACM conference on Recommender systems, 2007, pp. 25–32.
[19] N. Hurley, Z. Cheng, and M. Zhang, “Statistical attack detection,” in Proceedings of the third ACM conference on Recommender systems, 2009, pp. 149–156.
[20] S. Ray and A. Mahanti, “Strategies for effective shilling attacks against recommender systems,” in Privacy, Security, and Trust in KDD, Springer, 2009, pp. 111–125.
[21] B. Van Roy and X. Yan, “Manipulation robustness of collaborative filtering,” Management Science, vol. 56, no. 11, pp. 1911–1929, 2010.
[22] B. Van Roy and X. Yan, “Manipulation-resistant collaborative filtering systems,” in Proceedings of the third ACM conference on Recommender systems, 2009, pp. 165–172.
[23] R. Bhattacharjee and A. Goel, “Algorithms and incentives for robust ranking,” in Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, 2007, pp. 425–433.
[24] J. Cao, Z. Wu, B. Mao, and Y. Zhang, “Shilling attack detection utilizing semi-supervised learning method for collaborative recommender system,” World Wide Web, vol. 16, no. 5–6, pp. 729–748, 2013.
[25] S. K. Lam and J. Riedl, “Shilling recommender systems for fun and profit,” in Proceedings of the 13th international conference on World Wide Web, 2004, pp. 393–402.