Um Método para Detecção de *Bots* Sociais Baseado em Redes Neurais Convolucionais Aplicadas em Mensagens Textuais

Paulo A. Braz[1], **Ronaldo R. Goldschmidt**[1]

[1]Seção de Engenharia da Computação (SE/8) – Instituto Militar de Engenharia (IME)
22.290-270 – Rio de Janeiro – RJ – Brazil

{paulo.braz,ronaldo.rgold}@ime.eb.br

Abstract. *Currently, social networks are subject to social bots that perform malicious activities such as the dissemination of fake news. Some research to detect this type of malware is based on statistics extracted from the content of posted messages. Once the extraction of statistics may lead to information loss, this work aims to present experimental evidences that the use of original textual messages can improve detection accuracy. To this end, we propose a method that applies a convolutional neural network to identify suspicious messages. Good preliminary results with Twitter data indicate adequacy of the proposed method.*

Resumo. *Atualmente, as redes sociais estão sujeitas a ações de bots sociais que executam atividades maliciosas como a disseminação de notícias falsas. Algumas pesquisas voltadas à detecção desse tipo de malware se baseiam em estatísticas extraídas a partir do conteúdo das mensagens postadas. Como a extração de estatísticas pode ocasionar perda de informação, este trabalho tem como objetivo apresentar evidências experimentais de que o uso de textos originais das mensagens pode melhorar a precisão de detecção. Para tanto, propõe-se um método que aplica uma rede neural convolucional para identificar mensagens suspeitas. Resultados preliminares utilizando dados do Twitter se mostraram promissores, fornecendo indícios de adequação do método proposto.*

1. Introdução

As redes sociais têm figurado como um espaço cada vez mais relevante no expediente de obtenção e disseminação de informações via Internet. De acordo com [Badri Satya et al. 2016], bilhões de usuários utilizam diariamente as mídias sociais, compartilhando informações sobre suas vidas e expressando opiniões através de mensagens, *likes* e avaliações. Por exemplo, dos cerca de 3,7 bilhões de usuários atualmente conectados à Internet, 1,5 bilhão utiliza o Facebook e aproximadamente 360 milhões possuem conta no Twitter[1].

As redes sociais encontram-se suscetíveis a ações dos *bots* sociais, contas automatizadas capazes de interagir com humanos e produzir conteúdo automaticamente. Contas deste tipo podem ser utilizadas para manipulação de opiniões de usuários das redes sociais por meio de atividades maliciosas como disseminação de boatos e notícias falsas, adulteração ou mesmo concepção de estatísticas de percepção pública, dentre outras [Ratkiewicz et al. 2011].

[1]**Statista** - www.statista.com/statistics/

A detecção de *bots* sociais é a tarefa de identificar contas automatizadas e diante deste contexto, faz-se necessário detectar e mitigar os efeitos nocivos destas entidades. Neste cenário, diversos trabalhos de pesquisa em detecção de *bots* sociais seguem a abordagem baseada em atributos ou características das contas (do inglês, *feature-based approach*) [Ferrara et al. 2016]. Os atributos a que se refere o nome desta abordagem são informações que descrevem o comportamento das contas das redes sociais. Número de amigos ou seguidores de uma conta, quantidade de contas que cada conta segue, número de mensagens propagadas em uma janela de tempo, quantidade de caracteres na descrição do perfil da conta, quantidade de fotos postadas, dentre outros, são exemplos de atributos comportamentais utilizados na detecção de *bots* sociais [Lee et al. 2011], [Davis et al. 2016] e [Wang et al. 2016]. Diante disso, é válido salientar que a maioria desses trabalhos aplica técnicas de aprendizado de máquina para, a partir dos dados sobre as contas de uma rede social, buscar construir modelos de classificação binária que sejam capazes de identificar com a maior precisão possível quais delas são *bots* sociais [Lee et al. 2011], [Davis et al. 2016].

Além de informações comportamentais sobre as contas das redes sociais, alguns dos trabalhos da abordagem de detecção de *bots* sociais baseada em atributos também consideram dados estatísticos sobre os textos das mensagens propagadas pelas contas, tais como: quantidade de links nas mensagens, número de menções a terceiros via "@nomeusuário", etc [Yang et al. 2014] e [Lee et al. 2011]. Nesses trabalhos, os textos brutos[2] das mensagens são submetidos a uma etapa de engenharia de atributos que é responsável por extrair dados estatísticos de interesse. No entanto, como qualquer consolidação de dados pode ocasionar perda de informação [Beaudry and Renner 2012], o presente trabalho levanta como hipótese que, ao utilizarem estatísticas para representar os textos das mensagens, os trabalhos relacionados possam ter eliminado indícios úteis para uma detecção de *bots* sociais mais precisa.

Assim sendo, o objetivo deste artigo é apresentar um conjunto de evidências experimentais de que a utilização de textos brutos de mensagens postadas por contas de redes sociais pode contribuir para produzir modelos de detecção de *bots* sociais mais precisos do que aqueles que se baseiam exclusivamente nos atributos comportamentais dessas contas. Para tanto, o artigo propõe um método de detecção de *bots* sociais que utiliza uma rede neural convolucional[3] [LeCun et al. 2004] adaptada para analisar os textos brutos das mensagens a fim de identificar quais são suspeitas de terem sido produzidas por *bots*. Em seguida, aplica-se um modelo que busca classificar cada conta como *bot* ou não *bot* com base nos atributos comportamentais e no percentual de mensagens da conta em questão que tenham sido consideradas suspeitas pela etapa anterior. Os resultados obtidos em um experimento preliminar envolvendo dois subconjuntos de contas e mensagens do Twitter se mostraram promissores, fornecendo indícios de adequação do método proposto.

O restante do artigo segue organizado da seguinte forma: a seção 2 apresenta a formalização do método proposto, descrevendo cada uma das etapas envolvidas; a seção 3 exibe um detalhamento acerca do experimento realizado e dos resultados preliminares

[2]Neste artigo, a expressão "texto bruto"será utilizada para referenciar o conteúdo original de uma mensagem que não tenha sido submetido a nenhum tipo de pré-processamento ou de engenharia de atributos.

[3]Este tipo de rede pertence a uma classe de algoritmos de aprendizado de máquina capazes de identificar características complexas a partir de características mais simples [Bezerra 2016]

obtidos; e, por fim, a seção 4 expõe as considerações finais do trabalho e aponta para possíveis alternativas de trabalhos futuros.

2. Método Proposto

Seja *N* uma rede social representada por um grafo dirigido $G(V, E)$, onde cada vértice $v \in V$ representa uma conta (ou usuário) de N e cada aresta $e \in E$ corresponde a um par ordenado entre duas contas u e v, representado por $e = (u, v)$, que indica que u segue v. Suponha que cada conta u possui um conjunto de mensagens $M_u = \{m_{u,1}, m_{u,2}, ..., m_{u,u_k}\}$ postadas por ela em N. O conjunto de todas as mensagens de N é representado por $M_N = \bigcup_{i=1}^{|V|} M_{u_i}$, onde $|V|$ corresponde à cardinalidade de V. Considere ainda que cada conta u possui uma lista ordenada de atributos $(u.a_1, u.a_2, ..., u.a_r, u.c)$, sendo $u.a_i$ comportamentais e $u.c$ um atributo binário que informa se u é *bot* ou não. O método proposto possui quatro etapas conforme ilustrado na Figura 1. A descrição de cada uma delas encontra-se detalhada a seguir.

– Etapa 1 - Seleção de Mensagens e Contas - Processo responsável por construir três conjuntos que serão utilizados pelas demais etapas: V', M'_{Treino} e M'_{Teste}. Inicialmente, considera-se que $V' = \emptyset$, $M'_{Treino} = \emptyset$ e $M'_{Teste} = \emptyset$. A construção dos conjuntos é realizada da seguinte forma: executa-se uma seleção aleatória sem reposição de x_{treino} mensagens de M_N. Tais mensagens são armazenadas em M'_{Treino}. Para cada mensagem $m \in M'_{Treino}$, recupera-se a conta u tal que $m \in M_u$ e atualiza-se $V' = V' \cup \{u\}$. Após a formação de V' e M'_{Treino}, inicia-se a construção de M'_{Teste}. Para cada $u \in V'$, são selecionadas aleatoriamente x_{teste} mensagens $m \in M_N$ tais que $m \notin M'_{Treino}$. Tais mensagens são incluídas em M'_{Teste}. Desta forma, ao final do procedimento, tem-se que $M'_{Treino} \cap M'_{Teste} = \emptyset$. Cabe ressaltar que tanto x_{treino} quanto x_{teste} são variáveis definidas pelo usuário do processo. Em seguida, toda mensagem $m \in M'_{Treino} \cup M'_{Teste}$ é enriquecida com a classificação da conta u responsável por sua propagação, i. e., $u.c$. Tal informação será utilizada na etapa 2.

– Etapa 2 - Classificação de Mensagens - Esta etapa subdivide-se em dois passos. No primeiro, treina-se uma rede convolucional com o conjunto M'_{Treino}. Para tanto, a rede recebe como entrada o texto bruto de cada mensagem m, sendo a classificação da mensagem a saída desejada $u.c$ a ser aprendida pela rede. Uma vez concluído o treinamento da rede neural, o segundo passo consiste em aplicar o modelo aprendido pela rede a cada uma das mensagens contidas em M'_{Teste}, contabilizando, ao final, para cada conta u, o percentual de mensagens classificadas como sendo suspeitas de terem sido propagadas por *bots* sociais.

– Etapa 3 - Enriquecimento de V' - Esta etapa é responsável por acrecentar uma nova informação em cada uma das contas $u \in V'$. Assim, a lista ordenada de atributos que descreve u passa a ser $(u.a_1, u.a_2, ..., u.a_r, u.c, u.susp)$, onde $u.susp$ contém o percentual de mensagens de u classificadas como suspeitas na etapa anterior.

– Etapa 4 - Classificação de Contas - Dado um algoritmo de classificação S, esta etapa realiza um processo de validação cruzada com k conjuntos sobre V'. Este

processo consiste em dividir V' em k conjuntos de contas para, em seguida, realizar k iterações. Em cada iteração, um dos k conjuntos é utilizado como conjunto de teste e os $k-1$ restantes utilizados para treinamento de S. O desempenho do modelo de classificação construído a cada iteração é armazenado. O processo se repete até que todos os k conjuntos tenham sido utilizados uma vez como conjunto de teste. Ao final do processo, é obtido o desempenho médio dos k modelos gerados. Em todas as iterações do processo, os atributos $a_1, a_2, ..., a_r, susp$ foram fornecidos como entradas e o atributo c como saída de S.

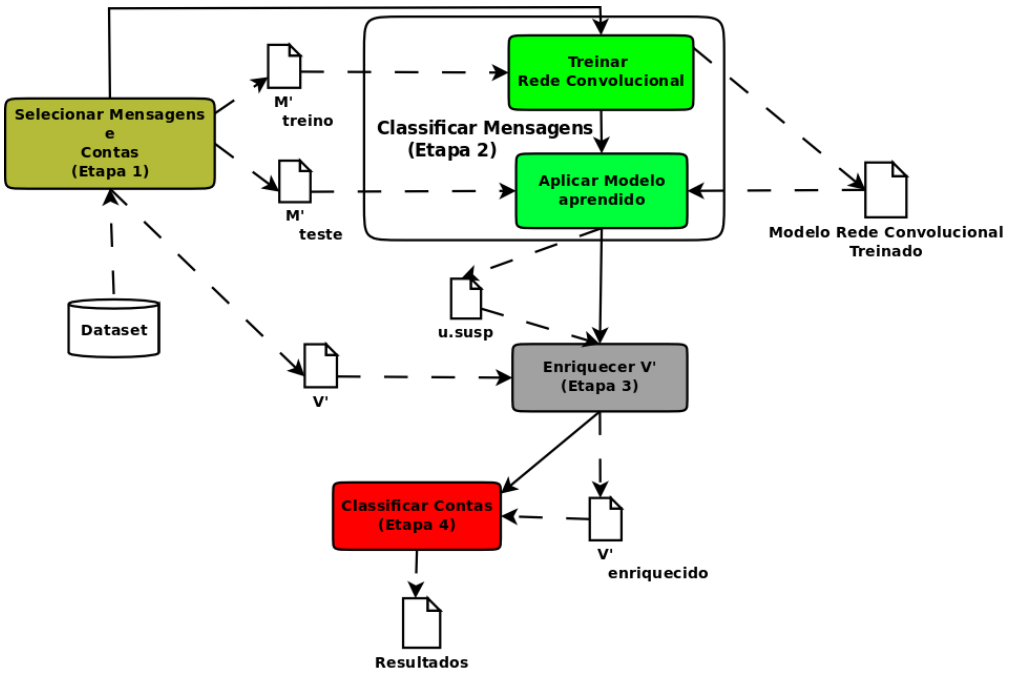

Figura 1. Visão Geral das Etapas do Método Proposto.

3. Experimento e Resultados Preliminares

A fim de avaliar o método proposto, foi realizado um experimento inicial com o mesmo *dataset* adotado em [Lee et al. 2011]. Trata-se de uma base de dados do *Twitter* com informações referentes ao período de dezembro de 2009 a agosto de 2010[4]. A base contém os atributos comportamentais indicados na Tabela 1, além dos textos de todas as mensagens publicadas no período. No total, o *dataset* contém aproximadamente 40 mil contas e 5,5 milhões postagens, sendo 87 postagens por conta, em média, aproximadamente. A escolha deste *dataset* deveu-se basicamente à disponibilidade dos atributos comportamentais em conjunto com os textos publicados pelas respectivas contas.

Diante do elevado volume de contas e mensagens disponíveis no *dataset* completo, o que demanda grande poder computacional e elevado tempo para o treinamento dos modelos de aprendizado de máquina, optou-se por utilizar subconjuntos menores na realização do treinamento do modelo para a geração dos resultados preliminares. Desta

[4]Neste *dataset*, cada conta contém um *label* que indica se a conta é falsa (*bot*) ou legítima.

Tabela 1. Atributos comportamentais disponíveis no *dataset* do experimento.

Atributos	Descrição
# usuários seguindo	número de usuários que a conta segue
#tweets	número de tweets postados pela conta
razão de #seguindo por #seguidores	razão de #usuários seguidos por #seguidores
#usuários seguidores	número de seguidores

forma, foram avaliados dois cenários: no primeiro foram utilizadas 2000 mensagens publicadas e no segundo 6000 mensagens para o treinamento e teste do classificador de mensagens. A seleção de mensagens e contas de ambos os cenários seguiu o procedimento descrito na etapa 1 do método proposto. A Tabela 2 apresenta um breve detalhamento estatístico sobre os dois cenários do experimento[5].

Tabela 2. Dados estatísticos sobre os cenários do experimento

	x_{treino}	x_{teste}	$x_{treino} + x_{teste}$	$\|V'\|$	Média Msg/Conta	Contas *bot*
Cenário 1	1000	1000	2000	53	37.7	50%
Cenário 2	3000	3000	6000	160	37.5	50%

A etapa de classificação de mensagens foi implementada em ambos os cenários com uma rede neural convolucional cuja configuração e arquitetura estão descritas na Tabela 3. A implementação deste modelo foi feita utilizando *Python* e *Keras* [Chollet et al. 2015].

Tabela 3. Arquitetura da rede convolucional

	Filtros	Kernel	Stride
2 Camadas	150	7X7	1
4 Camadas	150	3X3	1
Fully Connected	31 neurônios	–	–
Última camada	2 neurônios	–	–

As mensagens precisaram ser formatadas a fim de serem submetidas à rede na etapa 2. Para esta formatação foi utilizada a técnica *Char Quantization* proposta por [Zhang et al. 2015][6]. Desta forma, cada texto de mensagem publicada foi transformado em uma matriz binária de dimensão 150x64, uma vez que 150 é o limite máximo de caracteres para cada postagem no Twitter e 64 é a quantidade de caracteres considerados para mapeamento. A Figura 2 apresenta o conjunto de caracteres considerados para fins de mapeamento. Assim, as matrizes geradas foram esparsas. Cada linha de uma dada matriz correspondeu a um e, somente um, caracter da mensagem associada e, portanto, teve o *bit* setado para 1 exatamente na coluna relativa ao caracter identificado e 0 para as demais colunas da linha em questão.

[5]É válido ressaltar que foi adotada, em ambos os cenários, a mesma distribuição de classes encontradas no *dataset* original, em relação ao número de contas: 50% de contas do tipo *bot* e 50% de contas legítimas.

[6]A escolha por este tipo de formatação deveu-se basicamente pelo bom desempenho proporcionado por ela nos experimentos de classificação de texto relatados pelos autores.


```
['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v',
 'w', 'x', 'y', 'z', '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '-', ',', ';', '.', '!', '?', ':', '/',
 '\',',_', '@', '#', '$', '%', '^', '&', '*', '+', '-', '=', '<', '>', '(', ')', '[', ']', '{', '}']
```

Figura 2. Conjunto de caracteres usado para mapeamento das mensagens.

Ainda na etapa de Classificação de Mensagens foi executado o procedimento de *holdout* (com 2/3 de M'_{Treino} para treino e 1/3 de M'_{Treino} para validação), para cada um dos cenários. Deste modo, com o classificador treinado utilizando M'_{Treino}, fez-se a apresentação de M'_{Teste} para o modelo efetuar a classificação de cada mensagem m e, por conseguinte, viabilizar o cálculo do percentual de mensagens classificadas como sendo geradas por *bots* para todas as contas da amostra $(u.susp)$. Deste modo, com as taxas geradas, pode-se enriquecer V', finalizando assim a execução da Etapa 3.

Para classificação das contas foram utilizados dois algoritmos de classificação: *Random Forest* [Ho 1995] e MLP (*Multilayer Perceptron*) [Rosenblatt 1961]. Em cada cenário, cada algoritmo foi submetido a um processo de validação cruzada com 10 conjuntos. Os algoritmos executados foram instanciados com suas configurações *default*, utilizando Weka [Witten et al. 2016]. Importante notar que foi adotada a mesma parametrização dos algoritmos durante todo o processo. A Tabela 4 apresenta os parâmetros adotados por cada algoritmo.

Tabela 4. Configuração utilizada para cada algoritmo de classificação

Algoritmo	Configuração
MLP	hidden layers= 4, learning rate=0.3, momentum=0.2, epochs=500
Random Forest	bag size percent = 100, batch size = 100, num iterations = 100

A fim de comparar a influência dos textos brutos no processo de detecção de *bots* sociais, os mesmos algoritmos de classificação utilizados na etapa 4 foram aplicados sobre o conjunto V' sem o enriquecimento realizado pelo método proposto. Também neste caso, foi realizado um processo de validação cruzada com 10 conjuntos.

A Tabela 5 mostra as acurácias obtidas pelos dois algoritmos na classificação de contas em duas situações: uma com V' contendo apenas os atributos comportamentais da Tabela 1 e a outra com V' enriquecido pelo método proposto para conter também o percentual de mensagens suspeitas associado a cada conta. Pode-se perceber que, de uma forma geral, os algoritmos de classificação aplicados em V' enriquecido apresentaram desempenho superior aos mesmos algoritmos aplicados em V' sem enriquecimento, em ambos os cenários. Os resultados obtidos são, portanto, evidências experimentais de que a utilização de textos brutos de mensagens de contas de redes sociais pode contribuir para produzir modelos de detecção de *bots* mais precisos do que aqueles que se baseiam somente nos atributos comportamentais dessas contas. Ademais, aplicando os algoritmos de seleção de atributos RELIEF [Kononenko 1994] e CorrelationAttributeEval [Witten et al. 2016] (utiliza o coeficiente de Correlação de Pearson), verificou-se que o atributo criado pelo método proposto ficou em primeiro lugar no que se refere ao ganho de informação para a tarefa de detecção de *bots* em ambos os cenários. Este fato reforça a importância de se investigar alternativas de utilização de textos brutos das mensagens na construção de modelos de detecção de *bots* sociais.

Tabela 5. Resultados do experimento: acurácia dos classificadores

Cenário	Enriquecimento de V'	MLP			Random Forest		
		Acc.	FPs	FNs	Acc.	FPs	FNs
1	Não	69.23%	11%	19%	**86.5**%	7%	5%
1	Sim	84.61%	7%	7%	**90.3**%	5%	3%
2	Não	80.5%	13%	6%	**89.3**%	4%	6%
2	Sim	81.76%	8%	9%	**90.56**%	3%	5%

4. Considerações Finais

Atualmente as redes sociais encontram-se suscetíveis a ações dos *bots* sociais, contas automatizadas capazes de interagir com humanos e produzir conteúdo automaticamente. Contas deste tipo podem ser utilizadas para manipulação de opiniões de usuários das redes sociais por meio de atividades maliciosas como disseminação de notícias falsas e adulteração de estatísticas de percepção pública, dentre outras.

Algumas pesquisas voltadas à detecção de *bots* sociais se baseiam em estatísticas extraídas a partir do conteúdo das mensagens postadas. Como a extração de estatísticas pode ocasionar perda de informação, este trabalho teve como objetivo apresentar evidências experimentais de que o uso de textos originais das mensagens pode melhorar a precisão de detecção. Para tanto, propôs-se um método que aplica uma rede neural convolucional para identificar mensagens suspeitas. Baseado em atributos comportamentais e no percentual de mensagens suspeitas identificadas em cada conta, o método proposto aplica um classificador binário para detectar a presença de *bots* sociais. Resultados preliminares com dados do Twitter se mostraram promissores, fornecendo indícios de adequação do método proposto, assim como evidências experimentais de que a utilização de textos brutos de mensagens publicadas por contas de redes sociais pode contribuir para produzir modelos de detecção de *bots* sociais mais precisos do que aqueles que se baseiam apenas nos atributos comportamentais dessas contas.

Como trabalhos futuros, destacam-se a utilização de todo conteúdo textual e de todas as contas contidas no *dataset* completo adotado no experimento deste artigo, o emprego de testes estatísticos para avaliar a existência de diferença de desempenho significativa entre os classificadores e a utilização de outros *datasets* e algoritmos de classificação para avaliar a robustez do método proposto. Ainda seria oportuno avaliar a semântica de todas as mensagens propagadas. Ademais, seria interessante ainda avaliar a adequação de outras técnicas de aprendizado profundo para a classificação das mensagens, tais como redes neurais recorrentes do tipo LSTM [Hochreiter and Schmidhuber 1997], por exemplo.

Referências

Badri Satya, P. R., Lee, K., Lee, D., Tran, T., and Zhang, J. J. (2016). Uncovering fake likers in online social networks. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, pages 2365–2370, New York, NY, USA. ACM.

Beaudry, N. J. and Renner, R. (2012). An intuitive proof of the data processing inequality. *Quantum Info. Comput.*, 12(5-6):432–441.

Bezerra, E. (2016). Introdução à aprendizagem profunda. `http://sbbd2016.fpc.ufba.br/sbbd2016/minicursos/minicurso3.pdf`.

Chollet, F. et al. (2015). Keras. `https://github.com/fchollet/keras`.

Davis, C. A., Varol, O., Ferrara, E., Flammini, A., and Menczer, F. (2016). Botornot: A system to evaluate social bots. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 273–274. International World Wide Web Conferences Steering Committee.

Ferrara, E., Varol, O., Davis, C., Menczer, F., and Flammini, A. (2016). The rise of social bots. *Commun. ACM*, 59(7):96–104.

Ho, T. K. (1995). Random decision forests. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, volume 1, pages 278–282. IEEE.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Kononenko, I. (1994). Estimating attributes: analysis and extensions of relief. In *European conference on machine learning*, pages 171–182. Springer.

LeCun, Y., Huang, F. J., and Bottou, L. (2004). Learning methods for generic object recognition with invariance to pose and lighting. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–104. IEEE.

Lee, K., Eoff, B. D., and Caverlee, J. (2011). Seven months with the devils: A long-term study of content polluters on twitter.

Ratkiewicz, J., Conover, M., Meiss, M. R., Gonçalves, B., Flammini, A., and Menczer, F. (2011). Detecting and tracking political abuse in social media.

Rosenblatt, F. (1961). Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Technical report, CORNELL AERONAUTICAL LAB INC BUFFALO NY.

Wang, G., Zhang, X., Tang, S., Zheng, H., and Zhao, B. Y. (2016). Unsupervised clickstream clustering for user behavior analysis. In *SIGCHI Conference on Human Factors in Computing Systems*.

Witten, I. H., Frank, E., Hall, M. A., and Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Yang, Z., Wilson, C., Wang, X., Gao, T., Zhao, B. Y., and Dai, Y. (2014). Uncovering social network sybils in the wild. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 8(1):2.

Zhang, X., Zhao, J., and LeCun, Y. (2015). Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

