

# A Process for Building Datasets that Enable the Application of Different Methods for Detecting Fake News and Social Bots

Jeferson Luis Gonçalves, M.Sc\*

jefluis@ime.eb.br

Departamento de Ciência e Tecnologia  
Instituto Militar de Engenharia (IME)  
Rio de Janeiro, RJ, Brasil

Paulo Márcio Souza Freire, D.Sc

paulo.freire@dde.faetec.rj.gov.br

Fundação de Apoio à Escola Técnica do Estado do Rio de Janeiro (FAETEC)  
Rio de Janeiro, RJ, Brasil

Lucas Barboza de Menezes Torres

lucas.221014440007@aluno.eteot.faetec.rj.gov.br

Fundação de Apoio à Escola Técnica (FAETEC) - Escola  
Técnica Estadual Oscar Tenório (ETEOT)  
Rio de Janeiro, RJ, Brasil

Ronaldo Ribeiro Goldschmidt, D.Sc

ronaldo.rgold@ime.eb.br

Departamento de Ciência e Tecnologia  
Instituto Militar de Engenharia (IME)  
Rio de Janeiro, RJ, Brasil

## ABSTRACT

**Context:** The spread of fake news on social media is an imperative concern. The dissemination of such news by social bots has added complexity to disinformation detection applications. Machine learning methods have been applied to classify news as fake and not fake and accounts as bot and not bot, based on labeled datasets.

**Problem:** There are no datasets that embed simultaneously news labeled as fake and not fake and accounts labeled as bot and not bot. This gap hinders the evaluation of classification methods that could benefit from such data embedding.

**Solution:** A process for building datasets that contain pieces of news and accounts appropriately labeled, and enable development and comparison of fake news and social bots detection methods.

**IS Theory:** General Systems Theory<sup>1</sup> and social Network Theory<sup>2</sup>.

**Method:** Data requirements from SOTA<sup>3</sup> fake news and social bot detection methods guided the development of the process. This process collects data from social networks and fact-checking agencies. A case study generated a dataset, illustrating the viability of the process.

**Summary of Results:** The dataset generated is public and contains 440 labeled pieces of news and 6,274 labeled accounts. Most fake news detection methods improved their performance when they considered the labels of the accounts.

**Contributions and Impact on the IS field:** The process that builds datasets that integrate labeled news and labeled accounts, and the dataset generated by the case study. Both contributions are related to the Grand Challenges in IS Research and the Sociotechnical Vision of IS.

## CCS CONCEPTS

- Computing methodologies → Machine learning; Artificial intelligence;
- Information systems → Data mining; social networking sites; Web mining.

\*Both authors contributed equally to this research.

<sup>1</sup>focusing on the interaction between independent components within a system.

<sup>2</sup>represented by the integration of data from fake news and social bots.

<sup>3</sup>State-of-the-art

## KEYWORDS

Fake News, Social Bots, Social Networks, Datasets, Machine Learning

## 1 INTRODUÇÃO

Tradicionalmente, a veiculação de notícias se limitava aos meios de comunicação convencionais, como rádio, televisão, revistas e jornais impressos. No entanto, devido à facilidade de acesso e ao baixo custo, vem se destacando o uso dos meios digitais de divulgação de notícias (MDDN), que incluem as mídias virtuais (e.g.: jornais e revistas on-line), assim como as redes sociais virtuais (RSV) [13, 19].

No caso particular das RSV, algumas permitem que indivíduos, independentemente de sua credibilidade, divulguem notícias com grande potencial de propagação [3, 35]. Essa permissividade tem intensificado um problema antigo: a disseminação intencional de notícias falsas, as denominadas *fake news* [31].

Como exemplo recente desse tipo nocivo de notícia divulgada em RSV, pode-se destacar as *fake news* relacionadas com as fortes chuvas que assolararam mais de 400 cidades no Rio Grande do Sul<sup>4</sup>. Essas notícias falsas atrapalharam os trabalhos de assistência à população atingida pelas enchentes e foram utilizadas com o objetivo de causar desinformação.

Um aspecto que tem agravado a disseminação de *fake news* nas RSV é a divulgação de muitas delas por *bots sociais* [12, 32]. Um *bot social* é uma conta gerenciada por software, capaz de disseminar conteúdo de forma automática [12]. Embora muitos *bots sociais* sejam benignos, pode-se criar *bots malignos* com o objetivo de persuadir, difamar ou enganar [13].

Dante desse cenário, tanto a academia quanto a indústria têm pesquisado como combater *fake news* e *bots sociais* nas RSV. Contudo, esse problema é complexo devido ao volume de publicações e à velocidade de propagação. Assim, o uso de métodos computacionais tem se destacado devido à sua velocidade de atuação [12, 26]. Muitos desses métodos empregam técnicas de aprendizado de máquina para classificar notícias como *fake* e *não fake* [32] e contas como *bot* e *não bot* [12]. Para tanto, esses métodos precisam treinar, validar e testar modelos a partir de *datasets* rotulados.

<sup>4</sup><https://netlab.eco.ufrj.br/post/enchentes-norio-grande-do-sul-uma-análise-da-desinformação-multiplataforma-sobre-o-desastre-climáti>

Até onde foi possível observar, não foram encontrados *datasets* que contenham notícias rotuladas como *fake* e *não fake* e contas rotuladas como *bot* e *não bot*, de forma integrada. A carência de *datasets* com essas características dificulta pesquisas voltadas ao desenvolvimento e à avaliação de métodos de detecção que possam tirar proveito dessa integração. Por exemplo, um método que, ao classificar uma notícia como *fake* e *não fake*, possa utilizar, dentre outras informações presentes no *dataset*, o rótulo da conta divulgadora dessa notícia (i.e., se a referida conta divulgadora é rotulada como *bot* ou *não bot*). Para abordar a carência de *datasets* mencionada acima, o presente trabalho tem como objetivo e principal contribuição propor um processo de construção de *datasets* que contenham informações de notícias e contas, incluindo os seus respectivos rótulos. Esse processo foi proposto com base em requisitos levantados a partir de uma revisão bibliográfica sobre demandas de informações necessárias aos métodos de detecção de *fake news* e *bots sociais* encontrados na literatura.

Para ilustrar o potencial do processo proposto e de *datasets* que possam ser gerados a partir dele, foi realizado um estudo de caso, por meio da implementação de um protótipo. Para tanto, foram utilizadas a rede social X<sup>5</sup> (antigo Twitter), as agências de checagem de fatos Aos Fatos<sup>6</sup> e Lupa<sup>7</sup>, assim como os jornais on-line G1<sup>8</sup> e R7<sup>9</sup>. Diferentemente do que é possível realizar nos *datasets* gerados por outros processos de construção de *datasets* sobre *fake news*, no *dataset* gerado pelo protótipo implementado, foram aplicados métodos de detecção de *fake news* que, além das informações das notícias, utilizaram os rótulos das contas divulgadoras (*bot* ou *não bot*) dessas notícias. Ao considerar os rótulos dessas contas, a maioria desses métodos obteve um desempenho superior ao alcançado por eles sem utilizar tais informações.

Este artigo está organizado em mais cinco seções. A Seção 2 descreve os trabalhos relacionados, utilizando um modelo comparativo. O processo de construção de *datasets* é proposto na Seção 3. A Seção 4 detalha o estudo de caso, incluindo o protótipo, a descrição do *dataset* gerado e a análise dos resultados. Por fim, a Seção 5 apresenta as considerações finais, destacando os resultados obtidos e perspectivas para trabalhos futuros.

## 2 TRABALHOS RELACIONADOS

Dado o impacto significativo das *fake news* e *bots sociais* na disseminação de desinformação, torna-se fundamental compreender e avaliar os *datasets* utilizados para criação de métodos de detecção de *fake news* ou de *bots sociais*. Esta seção revisa trabalhos que consuem *datasets*, mesmo que o objetivo principal desses trabalhos seja apresentar métodos de detecção de *fake news* ou de *bots sociais*.

A fim de facilitar a análise desses trabalhos, foi proposto um modelo comparativo. Os critérios de comparação pertencentes a esse modelo foram estabelecidos a partir da análise das informações disponibilizadas pelos *datasets* presentes nos trabalhos do estado da arte pesquisados. Tais critérios estão apresentados na Tabela 1.

Até onde foi possível observar, foram identificados 34 *datasets* publicados a partir de 2018. Aplicando o modelo comparativo proposto

<sup>5</sup><https://x.com/>

<sup>6</sup><https://www.aosfatos.org/>

<sup>7</sup><https://lupa.uol.com.br/>

<sup>8</sup><https://g1.globo.com/>

<sup>9</sup><https://www.r7.com/>

**Tabela 1: Critérios de Comparação**

Critério	Descrição
AC (atributos de conta)	indica se o <i>dataset</i> tem informações da conta (e.g.: número de seguidores e data da criação)
RC (rotulação de conta)	exprime se há no <i>dataset</i> a informação do rótulo da conta (e.g.: <i>bot</i> ou <i>não bot</i> )
AN (atributos de notícia)	aponta se o <i>dataset</i> tem informações da notícia (e.g.: texto da notícia e título)
RN (rotulação de notícia)	sinaliza se há no <i>dataset</i> a informação do rótulo da notícia (e.g.: <i>fake</i> ou <i>não fake</i> )
MC (modelo de dados conceitual)	indica se o trabalho possui uma representação em níveis mais altos de abstração dos dados
PC (processo de construção)	revele se o trabalho demonstrou o processo utilizado para a construção do <i>dataset</i>

neste estudo, esses *datasets* são apresentados na Tabela 2. Desse conjunto, são brevemente descritos os quatro trabalhos mais recentes que demonstraram o processo de construção de *datasets*, sendo dois relacionados a *fake news* e dois a *bots sociais*. Mais detalhes podem ser consultados nas respectivas referências.

- **Bracis2019FakeNews** - Disponibiliza dois *datasets*<sup>10</sup> no idioma português. O primeiro contém textos de 115 notícias falsas e 12 verdadeiras divulgadas via WhatsApp. O segundo é composto por 87 identificadores de postagens falsas e 21 de postagens verdadeiras, coletadas da rede social X;
- **FakeNewsSetGen** - Contém dados de notícias das agências de checagem e de portais de notícias em língua portuguesa, totalizando 300 notícias *fake* e 300 notícias *não fake*. Inclusive, com dados enriquecidos das redes sociais que compartilharam tais notícias. O referido *dataset* se encontra no repositório do trabalho<sup>11</sup>;
- **SocialBots** - Apresenta um processo de construção de *datasets* utilizando somente atributos quantitativos das contas de usuários. O referido processo produziu uma instância de *dataset* com 4.000 contas e mais de 10 milhões de *tweets* com a temática das Olimpíadas de 2016;
- **SocialBotsDetection** - Trabalho de detecção de *bots sociais* que utiliza um léxico de sentimentos aplicado nas mensagens e apresenta um processo de construção de *datasets*. O primeiro *dataset* construído contém 238.455 notícias divulgadas por *bots* e 436.798 notícias divulgadas por *não bots*. O segundo possui 1.637 contas de *bots* e 2.981 contas de *não bots*.

Dentre os trabalhos apresentados, o presente estudo observou que o *FakeNewsSetGen* [6] foi o que se enquadrou na maioria dos critérios de comparação utilizados. Além disso, o *FakeNewsSetGen*

<sup>10</sup>[https://github.com/phfaustini/BRACIS2019\\_FAKENEWS](https://github.com/phfaustini/BRACIS2019_FAKENEWS)

<sup>11</sup><https://github.com/kamplus/FakeNewsSetGen>

**Tabela 2: Comparação entre os *Datasets* Relacionados**

<i>Dataset</i>	AC	RC	AN	RN	MC	PC
Bracis2019FakeNews [10]	-	-	X	X	-	X
BuzzFace [27]	X	-	X	X	-	-
DataSet Emergent [42]	-	-	X	X	-	-
DistrustRank [37]	X	-	X	X	-	-
Factck.Br Corpus [21]	-	-	X	X	-	X
FakeBr Corpus [20]	X	-	X	X	-	X
FakeNews vc Satire [15]	-	-	X	X	-	-
FakeNewsNet2 [30]	X	-	X	X	-	X
FakeTweet.Br Corpus [4]	-	-	X	X	-	X
Politicalnews [2]	-	-	X	X	-	-
PolitiFact para XFake [39]	X	-	X	X	-	-
Twitter e Sina Weibo para Early Detection [16]	X	-	X	X	-	-
Twitter e Sina Weibo para TCNN-URG [24]	-	-	X	X	-	-
Twitter para content Polluters [22]	X	-	X	X	-	-
Twitter para TraceMiner [38]	X	-	-	X	-	-
Twitter Trec [33]	-	-	-	X	-	-
FakeNewsSetGen [6]	X	-	X	X	X	X
HoaxyBots [29]	X	X	-	-	-	-
Astroturf [28]	X	X	X	-	-	-
TwiBot-20 [11]	-	X	X	-	-	-
German Bot [25]	-	X	-	-	-	-
Verified-2019 [41]	-	X	-	-	-	-
Botwiki-2019 [41]	-	X	-	-	-	-
Midterm-2018 [41]	-	X	-	-	-	-
Rtbust [18]	-	X	-	-	-	-
Political-bots [40]	-	X	-	-	-	-
Botometer-feedback [40]	-	X	-	-	-	-
Vendor-purchased [40]	-	X	-	-	-	-
Celebrity [40]	-	X	-	-	-	-
Pronbots [40]	-	X	-	-	-	-
Stock-bot Acitivity in Stock Microblos on Twitter [5]	X	X	-	-	-	-
SocialBots [7]	X	X	X	-	-	X
Loyola-González [17]	X	X	-	-	-	-
SocialBotsDetection [8]	X	X	X	-	-	X

foi o único que disponibilizou o modelo de dados conceitual, fato que facilitou o aprofundamento do seu entendimento.

Entretanto, não somente o *FakeNewsSetGen* como também os demais trabalhos pesquisados não propuseram a construção de *datasets* que contenham, ao mesmo tempo, tanto a rotulação da notícia (*fake* e *não fake*) como da conta (*bot* e *não bot*). A carência de *datasets* com essas características dificulta a realização de pesquisas relacionadas a métodos de detecção que possam tirar proveito dessa integração. Com base nessa lacuna, o presente trabalho tem como objetivo e principal contribuição propor um processo de construção de *datasets* que contenham informações das notícias e contas, incluindo os seus respectivos rótulos.

### 3 PROCESSO DE CONSTRUÇÃO

Esta seção tem como objetivo apresentar o *FakeNewsAndSocialBotSetGen*, processo de construção de *datasets* proposto pelo presente

trabalho. Sendo a principal contribuição desta pesquisa, o *FakeNewsAndSocialBotSetGen* é uma extensão do *FakeNewsSetGen*. O *FakeNewsSetGen* foi escolhido como o processo a ser estendido, pois, como já analisado, esse foi o trabalho que se enquadrou na maioria dos critérios de comparação utilizados na análise dos trabalhos relacionados. Diferentemente do *FakeNewsSetGen*, onde somente as notícias são rotuladas como *fake* e *não fake*, o *FakeNewsAndSocialBotSetGen* constrói *datasets* que contenham não só a rotulação das notícias em *fake* ou *não fake* como também a rotulação das contas em *bot* ou *não bot*.

Iniciando pelo levantamento de requisitos<sup>12</sup>, foram identificados os atributos dos *datasets* usados por métodos de detecção de *fake news* e *bots sociais* do estado da arte, bem como informações disponíveis em diferentes MDDN. Conforme descrito na Seção 3.1, esses

<sup>12</sup>Fase da engenharia de requisitos que busca levantar informações sobre o que o sistema deve fazer [23]

atributos foram estruturados e abstraídos em um modelo de dados conceitual projetado para suportar diferentes métodos de detecção de *fake news* e *bots sociais*.

O levantamento de requisitos ainda identificou funcionalidades presentes nos trabalhos relacionados, viabilizando a criação de um modelo macro-funcional para descrever conceitualmente o funcionamento do *FakeNewsAndSocialBotSetGen*. Como detalhado na Seção 3.2, esse modelo favorece o entendimento de como e quando ocorrem a coleta, o processamento e o armazenamento dos dados referentes aos conjuntos de atributos do modelo de dados conceitual. Cabe destacar que esse modelo macro-funcional descreve o processo de rotulação das notícias como *fake* e *não fake* e das contas divulgadoras dessas notícias como *bot* e *não bot*. No caso das notícias, fontes fidedignas (agências de checagem de notícias e mídias virtuais renomadas) fornecem essas notícias, incluindo a possibilidade de obtenção dos seus respectivos rótulos (*fake* e *não fake*). Entretanto, diferentemente dos rótulos das notícias, os rótulos das contas não se encontram disponíveis. Com base nessa lacuna de informação, para cada conta coletada são aplicados modelos de classificação de conta em *bot* e *não bot*, previamente construídos a partir de outros *datasets* contendo contas rotuladas. Por fim, para cada conta, é realizado um processo de avaliação de concordância entre as classificações obtidas, em busca de uma rotulação como *bot* ou *não bot*. Maiores detalhes sobre o processo proposto encontram-se expostos nas próximas subseções.

### 3.1 Modelo de Dados Conceitual

Ilustrado na Figura 1, o modelo de dados conceitual (MDC) do *FakeNewsAndSocialBotSetGen* explicita os conjuntos de atributos e as suas relações. Para apresentação desse modelo foi utilizada a notação de Diagrama de Classes de Análise da UML [1]. Tendo como base o MDC do *FakeNewsSetGen*, foram adicionadas as classes Modelo e Resultado da Classificação, visando abranger a rotulação das contas (*bot* e *não bot*). Ademais, a classe Conta foi expandida para incluir mais atributos, buscando atender às necessidades dos modelos de classificação de contas em *bot* e *não bot*. A seguir, é apresentada uma breve descrição das classes do MDC desenvolvido.

- Meio de Divulgação - Compreende os canais por onde as notícias podem ser divulgadas (Mídia Virtual e Rede social);
- Fonte Fidedigna - Corresponde às agências de checagem de fatos (e.g.: Lupa e Aos Fatos) ou às mídias virtuais renomadas (e.g.: portais de notícias G1 e R7);
- Notícia - Engloba as características da notícia obtida em uma fonte fidedigna e pode conter diferentes componentes (e.g.: texto, imagem e áudio), assim com armazena o seu respectivo rótulo (*fake* ou *não fake*);
- Divulgação - Representa todas as divulgações sofridas por uma determinada notícia ou desencadeadas a partir de outra divulgação (e.g.: *tweets* e *retweets*);
- Componente - Engloba atributos de diferentes tipos (e.g.: texto, imagem e áudio) que podem estar presentes em notícias e divulgações;
- Conta - Contém dados das contas de usuários de meios de divulgação (e.g.: número de seguidores e data da criação), incluindo o rótulo inferido da conta como *bot* ou *não bot*, conforme concordância de modelos de classificação de contas

como *bot* ou *não bot*, que tenham sido treinados previamente a partir de outros *datasets* contendo contas rotuladas com relação a sua classificação em *bot* e *não bot*;

- Proprietário - Compreende os dados do responsável por cada conta de usuário, sempre que esse responsável for identificado;
- Modelo - Engloba os modelos de classificação de contas utilizados para inferir o rótulo das contas (*bot* ou *não bot*);
- Resultado da classificação - Armazena a classe inferida por cada modelo de classificação de contas em *bot* e *não bot*, aplicado em cada conta.

### 3.2 Modelo Macro-Funcional

A Figura 2 apresenta o modelo macro-funcional do *FakeNewsAndSocialBotSetGen* (MMF). Os retângulos de cantos arredondados representam as etapas do processo e os retângulos de cantos retos representam as entidades externas. A seta tracejada indica uma relação de generalização entre as entidades externas e as setas contínuas indicam fluxos de dados. As Etapas destacadas 5 e 6 são extensões realizadas no modelo macro-funcional existente no *FakeNewsSetGen*.

Sempre que executado, o *FakeNewsAndSocialBotSetGen* permite gerar uma instância de *dataset* denominada *FakeNewsAndSocialBotSet*, compatível com o MDC apresentado anteriormente. Para tanto, o processo executa as etapas descritas a seguir.

A primeira etapa, detalhada pelo Algoritmo 1, recebe como parâmetro de entrada o conjunto  $S$  das fontes fidedignas a serem consultadas em busca de notícias previamente rotuladas. Cada fonte  $s \in S$  possui um atributo  $s.url$  que contém o endereço do feed de notícias de  $s$ . Assim, para cada  $s \in S$ , esta etapa recupera, a partir do feed  $s.url$ , o conjunto  $URLs$  de endereços de notícias previamente rotuladas por  $s$ . Em seguida, para cada  $url \in URLs$ , a notícia rotulada localizada em  $url$  é recuperada e armazenada em um conjunto de notícias rotuladas  $N^R$ , inicialmente vazio. Cada  $n^r \in N^R$  possui, dentre outros, os atributos  $n^r.id$  (i.e., o identificador da notícia  $n^r$ ) e  $n^r.text$  (i.e., o texto da notícia que corresponde ao  $n^r.componente.conteudo$ , onde  $n^r.componente.tipo = texto$ ). Por fim,  $N^R$  é armazenado no *FakeNewsAndSocialBotSet*.

---

#### Algoritmo 1: Identificar notícias rotuladas

---

```

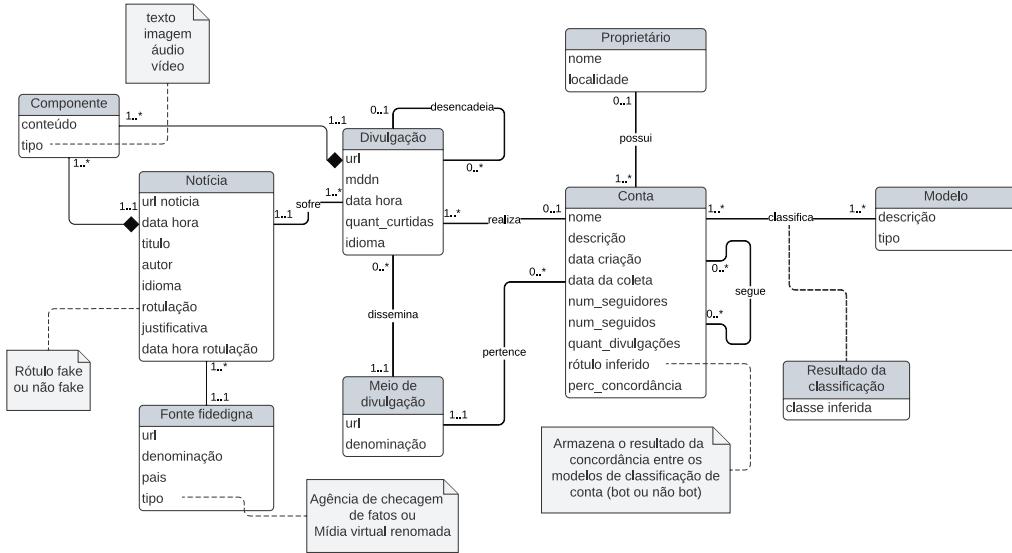
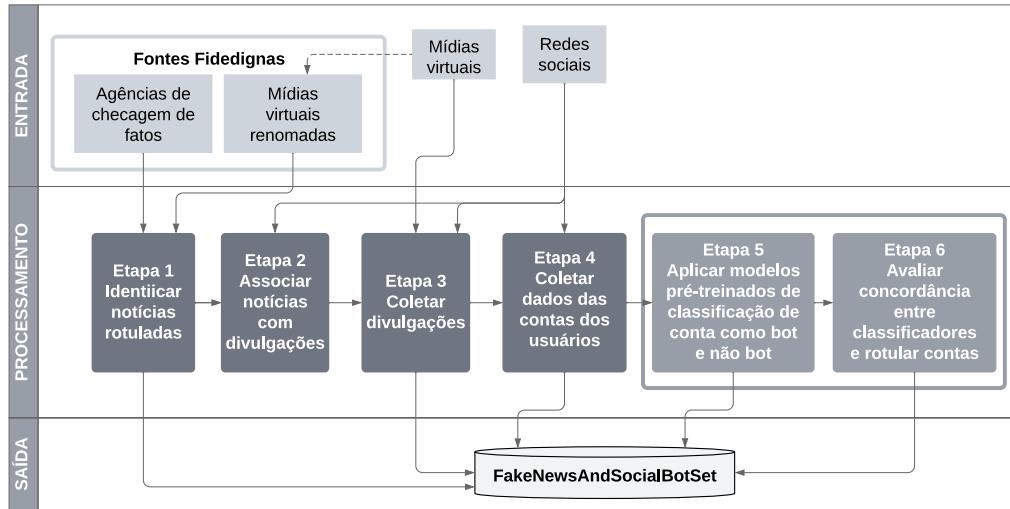
Entrada: Conjunto de fontes fidedignas selecionadas,  $S$ 
Saída: Conjunto de notícias rotuladas,  $N^R$ 
1 para cada  $s \in S$  faça
2   |   URLs  $\leftarrow$  ColetarURLsFeedRSS( $s.url$ );
3   |   para cada  $url \in URLs$  faça
4   |   |    $N^R \leftarrow N^R \cup$  lerNoticiaRotulada( $url$ );
5   |   fim
6 fim
7 Armazenar ( $N^R$ );
8 retorna  $N^R$ ;

```

---

Na segunda etapa, identificam-se os eventos de divulgação na rede social e nas mídias virtuais de cada notícia rotulada em  $N^R$ , conforme descrito no Algoritmo 2. Para cada  $n^r \in N^R$ , o texto  $n^r.text$  é pré-processado<sup>13</sup> a fim de se obter um conjunto de palavras-chave

<sup>13</sup>Pré-processamento envolve remoção de stop words e tokenização.

Figura 1: Modelo de Dados Conceitual (MDC) do *FakeNewsAndSocialBotSetGen*Figura 2: Modelo Macro-Funcional (MMF) do *FakeNewsAndSocialBotSetGen*.

*query* que é utilizado, juntamente com o intervalo de tempo de restrição da busca  $t$ , para identificar divulgações de  $n^r$  na rede social e nas mídias virtuais. As divulgações também são pré-processadas para facilitar a comparação textual e na função “PesquisarDivulgações” é realizada a verificação de inconsistências ou de informações incompletas na divulgação encontrada. Caso isto aconteça, essa divulgação não é coletada e a função passa para a próxima divulgação. As URLs das divulgações coletadas são armazenadas em  $URLsDIV$ .

Em seguida, cada  $urlDiv \in URLsDIV$  gera um par  $(n^r.id, n^r.urlDiv)$  que é armazenado em  $N^D$ .

A terceira etapa do processo, descrita no Algoritmo 3, coleta os detalhes das divulgações das notícias nas mídias virtuais ou redes sociais. Nessa etapa, para cada  $n^d \in N^D$ , são recuperados os dados de divulgação de  $n^d.urlDiv$  sobre a notícia  $n^d$ . Esses dados são armazenados no conjunto de divulgações  $D$ , onde, para cada  $d \in D$ ,  $d.mddn$  indica se a divulgação é originada de rede social ( $rsv$ ) ou

**Algoritmo 2:** Associar notícias com divulgações

**Entrada:** (1) Conjunto de notícias rotuladas,  $N^R$   
 (2) Intervalo de tempo a ser utilizado para restringir a busca por divulgações,  $t$

**Saída:** Conjunto de associações entre os identificadores de notícias e de divulgações,  $N^D$

---

```

1 para cada  $n^r \in N^R$  faça
2   query, URLsDIV  $\leftarrow \emptyset$  ;
3   query  $\leftarrow$  PreProcessar( $n^r.text$ ) ;
4   URLsDIV  $\leftarrow$  PesquisarDivulgacoes(query, t)
5   se URLsDIV  $\neq \emptyset$  então
6     para cada urlDiv  $\in URLsDIV$  faça
7       |  $N^D \leftarrow N^D \cup \{(n^r.id, urlDiv)\}$ 
8     fim
9   fim
10 fim
11 retorna  $N^D$ ;
```

---

de mídia virtual ( $mv$ ). Finalmente, o conjunto de divulgações  $D$  é armazenado no *FakeNewsAndSocialBotSet*.

**Algoritmo 3:** Coletar divulgações

**Entrada:** Conjunto de associações entre os identificadores de notícias e de divulgações,  $N^D$

**Saída:** Conjunto de divulgações de notícia,  $D$

---

```

1 para cada  $n^d \in N^D$  faça
2   |  $D \leftarrow D \cup$  Obter( $n^d.urlDIV$ );
3 fim
4 Armazenar(D);
5 retorna D;
```

---

Denominada “Coletar dados das contas dos usuários”, a quarta etapa, descrita pelo Algoritmo 4, recupera dados de usuários que divulgam as notícias nas RSV. Nessa etapa, cada divulgação  $d \in D$ , onde  $d.mddn = rsv$ , é processada para coletar os dados da conta  $c$  que fez a divulgação  $d$ , bem como dos seus seguidores e seguidos. Esses dados são armazenados no conjunto  $C$  e, posteriormente, no *FakeNewsAndSocialBotSet*.

**Algoritmo 4:** Coletar dados das contas dos usuários

**Entrada:** Conjunto de divulgações de notícia,  $D$

**Saída:** Conjunto de contas dos usuários divulgadores, incluindo os seus respectivos seguidores e seguidos,  $C$

---

```

1 para cada  $d \in D$  faça
2   se  $d.mddn = rsv$  então
3     |  $C \leftarrow C \cup$  Obter(d);
4   fim
5 fim
6 Armazenar(C);
7 retorna C;
```

---

A quinta etapa, resumida pelo Algoritmo 5, utiliza um conjunto  $M$  de modelos de classificação de contas como *bot* e *não bot*, construídos previamente a partir de outros *datasets* contendo contas rotuladas. Para tanto, sendo  $M \neq \emptyset$ , cada conta  $c \in C$  é classificada por cada modelo  $m \in M$ , onde os resultados são armazenados no conjunto de contas classificadas  $C^M$ . É importante observar que a escolha dos modelos a compor o conjunto  $M$  deve ser criteriosa e priorizar modelos que tenham apresentado previamente, em outros *datasets*, uma robustez significativa com relação à sua capacidade de distinguir corretamente contas *bot* de contas *não bot*.

**Algoritmo 5:** Aplicar modelos de classificação de conta como *bot* e *não bot*

**Entrada:**

- (1) Conjunto de modelos de classificação de contas previamente treinados,  $M$
- (2) Conjunto de contas de usuários,  $C$

**Saída:** Conjunto de classificações das contas pelos modelos previamente treinados,  $C^M$

---

```

1 para cada  $m \in M$  faça
2   para cada  $c \in C$  faça
3     |  $C^M \leftarrow C^M \cup$  Classificar(m, c);
4   fim
5 fim
6 Armazenar( $C^M$ );
7 retorna  $C^M$ ;
```

---

A sexta e última etapa, descrita no Algoritmo 6, avalia a concordância dos classificadores, visando rotular as contas como *bot* ou *não bot*. As classificações de cada conta  $c \in C$ , armazenadas em  $C^M$  (i.e.,  $C^{Mc}$ ), são analisadas para calcular o percentual de concordância. Para isto, a partir de  $C^{Mc}$ , a função “CalculaPercConcordancia” retorna, para cada  $c$ , o percentual de maior prevalência das classificações atribuídas a  $c$  pelos modelos armazenados em  $M$ . Em seguida, é avaliado se o percentual de concordância retornado é maior que o atribuído ao parâmetro *perc\_minimo\_concordancia*, que pode ser ajustado para o processo, objetivando obter a classe majoritária (*bot* ou *não bot*) para cada  $c$ . O percentual de concordância e a classe majoritária são atribuídos, respectivamente, a  $c.perc_concordancia$  e  $c.rotulo_inferido$  e, posteriormente, armazenados no *FakeNewsAndSocialBotSet*.

**Algoritmo 6:** Avaliar concordância entre classificadores e rotular contas

**Entrada:**

- (1) Conjunto de contas de usuário,  $C$
- (2) Conjunto de classificações das contas pelos modelos previamente treinados,  $C^M$
- (3) Percentual mínimo de concordância, *perc\_minimo\_concordancia*

**Saída:** Conjunto de contas com rotulação inferida,  $C$

---

```

1 para cada  $c \in C$  faça
2   |  $c.rotulo_inferido \leftarrow$  “semconcordancia”;
3   |  $c.perc_concordancia \leftarrow 0$ ;
4   |  $C^{Mc} \leftarrow$  filtro( $c, C^M$ );
5   |  $c.perc_concordancia \leftarrow$  CalculaPercConcordancia( $C^{Mc}$ );
6   se  $c.perc_concordancia > perc_minimo_concordancia$  então
7     | |  $c.rotulo_inferido \leftarrow$ 
8     | |  $ObterClassificacaoMajoritaria(C^{Mc})$ 
9 fim
10 Armazenar(C);
```

---

É importante salientar que o *FakeNewsAndSocialBotSet* pode ser continuamente atualizado, por meio de execuções reiteradas do *FakeNewsAndSocialBotSetGen*<sup>14</sup>. Isso permite a adição, ao *dataset*, de dados relacionados às novas notícias e contas rotuladas, além da inclusão de novos modelos de classificação de contas previamente treinados.

<sup>14</sup>Ao ser aplicado o processo proposto, é fundamental que sejam cumpridas as legislações vigentes relacionadas à ética em pesquisa e à proteção de dados.

## 4 ESTUDO DE CASO

### 4.1 Protótipo

Para ilustrar o potencial do processo proposto, foi customizado um protótipo do *FakeNewsAndSocialBotSetGen* para coletar notícias em português das agências de checagem Aos Fatos e Lupa, assim como das mídias virtuais G1 e R7. A escolha das agências de checagem e das mídias virtuais deveu-se, basicamente, a dois motivos: são populares e oferecem interface simples para acesso a notícias e suas rotulações como *fake* e *não fake*. O protótipo obtém divulgações dessas notícias na rede social X. Para tanto, o protótipo foi ajustado para se adequar às características e restrições tecnológicas (e.g.: uso de credenciais de acesso e limite no número de consultas permitido por sessão de acesso) impostas pelo uso da API dessa rede social. Essas fontes de coleta foram escolhas pontuais utilizadas neste protótipo, ressaltando que o processo *FakeNewsAndSocialBotSetGen* possibilita a sua aplicação em diferentes agências de checagem de fatos, mídias virtuais e redes sociais.

O protótipo<sup>15</sup> foi desenvolvido na linguagem *Python* e segue as etapas descritas na Seção 3.2. Para a execução destas etapas foi utilizada uma máquina virtual com processador Intel(R) Core(TM) i7 e 64GB de RAM. Dessa forma, notícias foram obtidas das fontes indicadas, incluindo as associações com os seus respectivos *tweets* na rede social X, sendo agregadas informações sobre *retweets*, contas de usuários divulgadores, seguidores e seguidos. Com o objetivo de rotular as contas coletadas, foram aplicados 18 modelos de classificação de conta em *bot* ou *não bot* previamente treinados em outros *datasets*, a partir do método descrito em [8]. Na busca pela concordância entre as classificações fornecidas por esses modelos, foi imputado o valor de 66% ao parâmetro *perc\_minimo\_concordancia*, ou seja, o rótulo final somente foi atribuído a uma conta, se o percentual de concordância dos modelos em relação à classe majoritária (i.e., a classe mais indicada pelos modelos) fosse superior a 66%. Isso significa que, se mais de 2/3 dos modelos (i.e., 12 dentre os 18 modelos) concordassem com uma determinada classificação, essa classificação era atribuída ao rótulo final da conta (*bot* ou *não bot*).

A escolha dos modelos de classificação de contas em *bot* e *não bot* deveu-se fundamentalmente ao fato de que todos eles obtiveram acurácia superiores a 94% em *datasets* públicos, balanceados e voltados à criação de métodos de detecção de *bots sociais*. O código fonte desses modelos, assim como os *links* para os referidos *datasets* podem ser obtidos no já mencionado *github* deste protótipo.

Em resumo, os métodos de detecção de *bots sociais* propostos em [8] se baseiam em informações sobre as contas (i.e., data de criação, número de seguidores e seguidos, quantidade de divulgações, quantidade de caracteres existentes no nome e na descrição das contas) e na análise dos sentimentos evocados a partir dos textos divulgados por essas contas. A análise de sentimento de cada conta foi retratada por meio das representações vetoriais (*embeddings*) das palavras usadas nos referidos textos divulgados por essas contas. Essas representações vetoriais de cada palavra foram obtidas através da aplicação de duas abordagens distintas. Na primeira foi gerada uma representação vetorial para cada palavra dos textos, a partir dos valores de 3 dimensões: *Valence*, *Arousal* e *Dominance*.

Para tanto, foi utilizado o léxico de sentimento VAD [36]. Na segunda abordagem, a representação vetorial de cada palavra dos textos foi obtida a partir dos valores das 50 dimensões geradas por um modelo de linguagem específico para análise de sentimento (SSWE) [34]. Maiores detalhes sobre essas representações vetoriais podem ser obtidos em [8].

Para geração dos modelos de classificação de conta, foram utilizados seis algoritmos de classificação (Regressão Logística [14], *K-NN* [14], Árvore de Decisão [14], *Random Forest* [14], *Gradient Boosting* [14] e Redes Neurais [14]), variando três conjuntos de informações de entrada (informações sobre a conta, representação vetorial obtida via VAD e representação vetorial obtida via SSWE), resultando em um conjunto *M* composto por 18 modelos de classificação treinados previamente. A Tabela 3 resume os modelos de classificação de contas utilizados nos experimentos e as informações utilizadas na geração de cada um deles. Em todos os modelos, as informações usadas foram extraídas das classes “Notícia”, “Divulgação” e “Conta” do MDC descrito na Subseção 3.1.

Um fator adicional importante é o tempo de treinamento, que varia significativamente entre os modelos. Enquanto o Árvore de Decisão e o *K-NN* são os mais rápidos, com tempos de treinamento variando de 11 a 120 segundos, o *Gradient Boosting* e o *Random Forest* apresentam tempos de treinamento muito mais elevados, chegando a mais de 6.000 segundos, em algumas variações. Isso torna esses dois últimos modelos mais custosos em termos computacionais, o que pode ser uma limitação em ambientes com poucos recursos e grandes volumes de dados a serem processados. A Regressão Logística apresentou tempos de treinamento intermediários, em torno de 2.500 segundos.

**Tabela 3: Informações utilizadas nos modelos de *M* para classificação de contas**

Modelo de Classificação	Informações de entrada
Regressão Logística 1	Informações sobre a conta + VAD
Regressão Logística 2	Informações sobre a conta + SSWE
Regressão Logística 3	Informações sobre a conta + VAD + SSWE
<i>K-NN</i> 1	Informações sobre a conta + VAD
<i>K-NN</i> 2	Informações sobre a conta + SSWE
<i>K-NN</i> 3	Informações sobre a conta + VAD + SSWE
Árvore de Decisão 1	Informações sobre a conta + VAD
Árvore de Decisão 2	Informações sobre a conta + VAD
Árvore de Decisão 3	Informações sobre a conta + SSWE
Random Forest 1	Informações sobre a conta + VAD + SSWE
Random Forest 2	Informações sobre a conta + SSWE
Random Forest 3	Informações sobre a conta + VAD + SSWE
<i>Gradient Boosting</i> 1	Informações sobre a conta + VAD
<i>Gradient Boosting</i> 2	Informações sobre a conta + SSWE
<i>Gradient Boosting</i> 3	Informações sobre a conta + VAD + SSWE
Rede Neural 1	Informações sobre a conta + VAD
Rede Neural 2	Informações sobre a conta + SSWE
Rede Neural 3	Informações sobre a conta + VAD + SSWE

#### Informações sobre a conta:

data criação, número de seguidores, número de seguidos, qtd de divulgações, qtd de caracteres existentes no nome e na descrição

A partir da aplicação desses 18 modelos de classificação de contas em *bots* e *não bots*, previamente treinados (*M*) foram geradas

<sup>15</sup><https://github.com/jefluis/FakeNewsAndSocialBotSetGen>

18 classificações para cada conta. Por meio dessas classificações, foram determinados o percentual de concordância entre os classificadores, como também a classe majoritária a ser atribuída como rótulo para cada conta (*bot ou não bot*). Todos os dados coletados foram persistidos, resultando na instância de *dataset* denominada de *FakeNewsAndSocialBotSet*.

## 4.2 Dataset Gerado

A Tabela 4 apresenta um resumo estatístico da instância *FakeNewsAndSocialBotSet* sob a perspectiva das notícias. Destaca-se a elevada diferença entre os totais de divulgações de notícias *fake* e *não fake*. Essa desigualdade, também identificada no estudo *FakeNews-Tracker* [30], reforça a tendência observada em diferentes estudos de que as *fake news* têm maior potencial de propagação do que as notícias *não fake* [26].

**Tabela 4: Estatística da Instância *FakeNewsAndSocialBotSet* - Visão Notícias**

Informação analisada	<i>fake</i>	<i>não fake</i>
Notícias	220	220
Divulgações (Tweets)	70.073	1.935
Para todas as 440 notícias		
Contas de usuários divulgadores	6.274	
Média de seguidores por conta	19.366	
Média de seguidos por conta	2.264	
Média de notícias por conta	3,3	

A Tabela 5 apresenta dados estatísticos sob a perspectiva das contas divulgadoras de notícias. Os dados reforçam a ideia de que contas *bots* desempenham um papel crucial na disseminação de *fake news*, conforme evidenciado pela relação entre as divulgações de notícias *fake* (DNF) e a quantidade de notícias *fake* (QNF). Contas *bots* apresentam uma média de 278 divulgações por notícia *fake*, enquanto contas *não bot* têm uma média de 121 divulgações. Esse padrão aponta para a eficácia dos *bots* na amplificação de *fake news*.

Adicionalmente, ao comparar os resultados das relações DNF/QNF e DNNF/QNNF para contas *bots*, observa-se que contas *bots* divulgam, em média, 278 vezes notícias *fake*, enquanto divulgam apenas 4,8 vezes notícias *não fake*. Outro dado relevante é que contas *não bot* possuem, em média, 5,77 vezes mais seguidores que contas *bots*, sugerindo que contas *não bot* estão mais focadas em interações sociais, ao invés de apenas ter uma conta com o único objetivo de postar notícias.

Além das médias de divulgações por notícia, o número total de divulgações também aponta para uma atuação mais intensa das contas *bots* na disseminação de *fake news*. Contas *bots* estão associadas a 48.720 divulgações de notícias *fake*, número mais do que o dobro das 21.353 divulgações feitas por contas *não bot*. Esses dados indicam um forte envolvimento de *bots* na disseminação de *fake news*, sugerindo que *bots* não só amplificam a desinformação, mas também o fazem em uma escala muito maior que as contas *não bot*. Vale destacar que uma mesma notícia pode ser divulgada tanto por contas *bots* quanto por contas *não bot*.

**Tabela 5: Estatística da Instância *FakeNewsAndSocialBotSet* - Visão Conta**

Informação analisada	<i>bot</i>	<i>não bot</i>
Contas de usuários divulgadores	4.388	1.886
Notícias ( <i>fake</i> e <i>não fake</i> )	367	386
QNF = Quantidade de notícias <i>fake</i>	175	176
QNNF = Quantidade de notícias <i>não fake</i>	192	210
DNF = Divulgações Notícias <i>fake</i> ( <i>Tweets</i> e <i>Retweets</i> )	48.720	21.353
DNF / QNF	278	121
DNNF = Divulgações Notícias <i>não fake</i> ( <i>Tweets</i> e <i>Retweets</i> )	926	1009
DNNF / QNNF	4,8	4,8
Média de seguidores por conta	8.007	46.213
Média de seguidos por conta	2.175	3.004

## 4.3 Avaliação

Apesar do objetivo principal desse estudo não ser propor e avaliar métodos de detecção de *fake news*, visando ilustrar o potencial de uso do processo *FakeNewsAndSocialBotSetGen*, esta subseção apresenta os resultados obtidos com o *dataset* gerado (instância do *FakeNewsAndSocialBotSet*) ao serem utilizados alguns métodos de detecção de *fake news*. Para tanto, foram aplicados cinco dos métodos de detecção de *fake news* empregados em [32], onde cada um deles utilizou, respectivamente, modelos de classificação gerados a partir dos algoritmos Regressão Logística, *K-NN*, Árvore de Decisão, *Random Forest* e *Gradient Boosting*. Esses cinco métodos foram experimentados duas vezes, variando os atributos utilizados. Na primeira experimentação, foram utilizados somente os atributos da notícia (i.e., quantidade de *tweets*, *retweets* e curtidas) que são obtidos a partir das classes Notícia e Divulgação. A segunda experimentação, além dos já descritos atributos pertencentes à notícia, utilizou o atributo rótulo da(s) conta(s) divulgadora(s) (*bot* ou *não bot*), da classe Conta. Nesse segundo caso, se uma notícia *n* for divulgada por várias contas, o método utiliza o rótulo majotitário (*bot* ou *não bot*) obtido a partir dos rótulos das contas que divulgaram *n*. Caso *n* tenha sido divulgada por um número igual de contas rotuladas como *bot* e *não bot*, *n* é considerada como divulgada por *bot*. Em todas as experimentações foi realizada validação cruzada com 10 conjuntos.

A Tabela 6 apresenta os resultados dos experimentos realizados com o *FakeNewsAndSocialBotSet*, onde podem ser comparadas as métricas obtidas, por cada modelo de classificação de notícias *fake* ou *não fake*, com e sem o uso do atributo rótulo da conta (*bot* ou *não bot*). Como pode ser observado na referida tabela, foram calculadas as médias de métricas clássicas em Aprendizado de Máquina, como acurácia, precisão, *recall* e F1 [9]. Tais médias (e respectivos desvios-padrão) foram obtidas pelos modelos durante o processo de validação cruzada com 10 conjuntos. Quanto mais próxima a média de cada métrica for de 1 (ou 100%), mais robustos foram os modelos gerados para identificar *fake news*. É importante comentar também que, como retratado pelos baixos valores de desvio-padrão, houve pouca variação de desempenho entre os modelos gerados por cada algoritmo de classificação durante o processo de validação cruzada.

**Tabela 6: Comparação dos Resultados Obtidos**

Modelo	FakeNewsAndSocialBotSet sem utilização do atributo rotulo conta				FakeNewsAndSocialBotSet com utilização do atributo rótulo conta			
	Acurácia	Precisão	Recall	F1	Acurácia	Precisão	Recall	F1
Regressão Logística	<b>0.571 ± 0.060</b>	<b>0.644 ± 0.105</b>	<b>0.570 ± 0.056</b>	<b>0.512 ± 0.074</b>	0.560 ± 0.043	0.622 ± 0.095	0.559 ± 0.039	0.508 ± 0.043
K-NN	0.639 ± 0.074	0.646 ± 0.078	0.639 ± 0.074	0.636 ± 0.074	<b>0.704 ± 0.067</b>	<b>0.711 ± 0.066</b>	<b>0.705 ± 0.066</b>	<b>0.702 ± 0.067</b>
Árvore de Decisão	0.753 ± 0.066	<b>0.760 ± 0.064</b>	0.756 ± 0.070	0.752 ± 0.066	<b>0.758 ± 0.060</b>	0.758 ± 0.064	<b>0.764 ± 0.069</b>	<b>0.757 ± 0.060</b>
Random Forest	0.786 ± 0.065	0.795 ± 0.075	0.792 ± 0.075	0.781 ± 0.070	<b>0.794 ± 0.061</b>	<b>0.805 ± 0.063</b>	<b>0.798 ± 0.069</b>	<b>0.790 ± 0.060</b>
Gradient Boosting	0.766 ± 0.067	0.773 ± 0.070	0.767 ± 0.068	0.765 ± 0.068	<b>0.791 ± 0.070</b>	<b>0.797 ± 0.070</b>	<b>0.792 ± 0.071</b>	<b>0.790 ± 0.071</b>

A análise comparativa dos métodos de detecção de *fake news*, aplicada ao conjunto de dados *FakeNewsAndSocialBotSet*, revelou o impacto da inclusão do atributo rótulo da conta (*bot* ou *não bot*) nos resultados de classificação. Métodos com modelos gerados a partir dos algoritmos *K-NN*, Árvore de Decisão, *Random Forest* e *Gradient Boosting* apresentaram melhorias em métricas como acurácia, precisão, recall e F1, com a adição do referido atributo rótulo da conta. Especificamente, os modelos gerados a partir dos algoritmos *Random Forest*, *Gradient Boosting* e *K-NN* mostraram um aumento na acurácia de, respectivamente, 1,02%, 3,27% e 10,17%. Esses aumentos demonstram indícios da eficácia da inclusão desse atributo no aprimoramento da capacidade preditiva. Em contraste, o modelo gerado a partir do algoritmo de Regressão Logística não apresentou melhorias em suas métricas, sugerindo que esse modelo é menos sensível à inclusão do atributo rótulo da conta.

Os resultados apresentam indícios de que a incorporação do atributo referente ao rótulo da conta pode beneficiar certos algoritmos, especialmente àqueles baseados em árvores. No entanto, a eficácia dessa abordagem pode variar dependendo do modelo, enfatizando a necessidade de uma análise cuidadosa ao selecionar atributos para diferentes arquiteturas de aprendizado de máquina. Por fim, os resultados obtidos ilustram que o processo *FakeNewsAndSocialBotSetGen* propicia a criação de *datasets*, contendo notícias rotuladas como *fake* e *não fake* e contas rotuladas como *bot* e *não bot*, a serem utilizados por métodos de detecção que possam tirar proveito dessa integração.

## 5 CONSIDERAÇÕES FINAIS

A detecção de *fake news* e *bots sociais* nos MDDN não é trivial, devido ao volume e à velocidade das divulgações. Assim, vários métodos baseados em aprendizado de máquina têm sido propostos. Embora alguns desses métodos utilizem dados de notícias e de contas em redes sociais, até onde foi possível observar, nenhum *dataset* contém a informação se a notícia é *fake* ou *não fake* e se a conta é *bot* ou *não bot*, de forma integrada.

Com base no exposto, o objetivo principal deste trabalho foi propor o processo *FakeNewsAndSocialBotSetGen*, que constrói *datasets* contendo dados sobre notícias rotuladas como *fake* ou *não fake* e os dados das contas divulgadoras rotuladas como *bot* ou *não bot*. Para ilustrar a aplicabilidade do processo proposto, foi realizado um estudo de caso que abrange a implementação de um protótipo que gerou o *dataset* *FakeNewsAndSocialBotSet*. Em seguida, métodos de detecção de *fake news* foram aplicados a esse *dataset*, cujos resultados foram comparados, apresentando indícios do potencial do processo e do *dataset* construído.

Como principal contribuição deste estudo, o processo *FakeNewsAndSocialBotSetGen* está relacionado aos "Grandes Desafios da Pesquisa em Sistemas de Informação no Brasil", ao propiciar a criação de *datasets* a serem aplicados na pesquisa e no ensino em Ciência de Dados.

Baseado nas limitações deste trabalho, podem ser considerados como trabalhos futuros: a implementação de interfaces com outras RSV para avaliar a aplicabilidade do processo em diferentes contextos; a adição de outras fontes fidedignas (agências de checagem de notícias e mídias virtuais renomadas); a avaliação de possíveis vieses na rotulação das contas; o aperfeiçoamento do processo de recuperação de divulgações de notícias nas RSV; a implementação de novos modelos de classificação de contas; a realização de experimentos com métodos de detecção de *bots*, com e sem utilização do rótulo das notícias (*fake* e *não fake*); assim como a apresentação de uma análise comparativa entre o *FakeNewsAndSocialBotSetGen* e outros processos de construção de *datasets*. Também sugere-se como trabalho futuro a variação do percentual mínimo de concordância entre os modelos de classificação de contas, buscando tentar identificar uma correlação desse percentual com o desempenho dos modelos de classificação de notícias.

## REFERÊNCIAS

- [1] Eduardo Bezerra. 2018. *Princípios de Análise e Projeto de Sistemas com UML*.
- [2] Sonia Castelo, Thais Almeida, Anas Elghafari, Aécio Santos, Kien Pham, Eduardo Nakamura, and Juliana Freire. 2019. A topic-agnostic approach for identifying fake news pages. In *Companion proceedings of the 2019 World Wide Web conference*. 975–980.
- [3] Argus Antonio Barbosa Cavalcante, Paulo Márcio Souza Freire, Ronaldo Ribeiro Goldschmidt, and Claudia Marcela Justel. 2024. Early detection of fake news on virtual social networks: A time-aware approach based on crowd signals. *Expert Systems with Applications* 247 (2024), 123350. <https://doi.org/10.1016/j.eswa.2024.123350>
- [4] Paulo Roberto Cordeiro and Vladi Matias Pinheiro. 2019. Um corpus de notícias falsas do twitter e verificação automática de rumores em língua portuguesa. In *Proceedings of the Symposium in Information and Human Language Technology*. 219–228.
- [5] Stefano Cresci, Fabrizio Lillo, Daniele Regoli, Serena Tardelli, and Maurizio Tesconi. 2019. Cashtag Piggybacking: Uncovering Spam and Bot Activity in Stock Microblogs on Twitter. *ACM Transactions on the Web* 13, 2 (April 2019), 1–27. <https://doi.org/10.1145/3313184>
- [6] Flávio Roberto Matias da Silva, Paulo Márcio Souza Freire, Marcelo Pereira de Souza, Gustavo de A. B. Plenamente, and Ronaldo Ribeiro Goldschmidt. 2020. *FakeNewsSetGen - a Process to Build Datasets that Support Comparison Among Fake News Detection Methods*. In *Anais do XXVI Simpósio Brasileiro de Sistemas Multimídia e Web* (16 ed.) (Evento Online). SBC, Porto Alegre, RS, Brasil, 188–195. <https://sol.sbc.org.br/index.php/webmedia/article/view/13678> 2022.
- [7] Carla Chrytina de Castro Pacheco Ferreira. 2018. *Detecção de socialbots em redes sociais baseada em atributos quantitativos*. Ph.D. Dissertation. Instituto Militar de Engenharia, Rio de Janeiro.
- [8] Samir de O. Ramos., Ronaldo R. Goldschmidt., and Alex de V. Garcia. 2022. Social Bots Detection: A Method based on a Sentiment Lexicon Learned from Messages. In *Proceedings of the 24th International Conference on Enterprise Information Systems - Volume 1: ICEIS*. INSTICC, SciTePress, 273–280. <https://doi.org/10.5220/0011115000003179>

- [9] K. Faceli. 2011. *Inteligência artificial: uma abordagem de aprendizado de máquina*. Grupo Gen - LTC. <https://books.google.com.br/books?id=4DweIAEACAAJ>
- [10] Pedro Faustini and Thiago Covões. 2019. Fake News Detection Using One-Class Classification. In *2019 8th Brazilian Conference on Intelligent Systems (BRACIS)*. 592–597. <https://doi.org/10.1109/BRACIS.2019.00109>
- [11] Shangbin Feng, Herun Wan, Ningnan Wang, Jundong Li, and Minnan Luo. 2021. TwiBot-20: A Comprehensive Twitter Bot Detection Benchmark. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management* (Virtual Event, Queensland, Australia) (CIKM '21). Association for Computing Machinery, New York, NY, USA, 4485–4494. <https://doi.org/10.1145/3459637.3482019>
- [12] Emilio Ferrara, Onur Varol, Clayton Davis, Filippo Menczer, and Alessandro Flammini. 2016. The rise of social bots. *Commun. ACM* 59, 7 (jun 2016), 96–104. <https://doi.org/10.1145/2818717>
- [13] Paulo Márcio Souza Freire and Ronaldo Ribeiro Goldschmidt. 2019. Uma Introdução ao Combate Automático à Fake News em Redes Sociais Virtuais. In *Tópicos em Gerenciamento de Dados e Informações, SBDI* (2019 ed.) (Fortaleza, CE, Brazil). SBC, Fortaleza, CE, Brasi, 38–67. <https://sol.sbc.org.br/livros/index.php/sbc/catalog/view/62/275/517-1>
- [14] Aurélien Géron. 2019. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems* (2nd ed.). O'Reilly Media.
- [15] Jennifer Golbeck, Matthew Mauriello, Brooke Auxier, Keval H Bhanushali, Christopher Bonk, Mohamed Amine Bouzaghrane, Cody Buntain, Riya Chanduka, Paul Cheakalos, Jenninne B Everett, et al. 2018. Fake news vs satire: A dataset and analysis. In *Proceedings of the 10th ACM conference on web science*. 17–21.
- [16] Yang Liu and Yi-Fang Brook Wu. 2018. Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence* (New Orleans, Louisiana, USA) (AAAI'18/IAAI'18/EAAI'18). AAAI Press, Article 44, 8 pages.
- [17] Octavio Loyola-González, Raúl Monroy, Jorge Rodríguez, Armando López-Cuevas, and Javier Israel Mata-Sánchez. 2019. Contrast Pattern-Based Classification for Bot Detection on Twitter. *IEEE Access* 7 (2019), 45800–45817. <https://doi.org/10.1109/ACCESS.2019.2904220>
- [18] Michele Mazza, Stefano Cresci, Marco Avvenuti, Walter Quattrociocchi, and Maurizio Tesconi. 2019. Rtbust: Exploiting temporal patterns for botnet detection on twitter. In *Proceedings of the 10th ACM conference on web science*. 183–192.
- [19] Uriel Merola, Paulo Freire, Ronaldo Goldschmidt, and Jorge Soares. 2023. Métodos de Detecção de Fake News: Uma Comparação entre as Abordagens de Crowd Signals e Ensembles. *Anais do Simpósio Brasileiro de Banco de Dados (SBDI)* (2023), 372–377. <https://doi.org/10.5753/sbdi.2023.233398>
- [20] Rafael A Monteiro, Roney LS Santos, Thiago AS Pardo, Tiago A De Almeida, Evandro ES Ruiz, and Oto A Vale. 2018. Contributions to the study of fake news in portuguese: New corpus and automatic detection results. In *Computational Processing of the Portuguese Language: 13th International Conference, PROPOR 2018, Canela, Brazil, September 24–26, 2018. Proceedings 13*. Springer, 324–334.
- [21] João Moreno and Graça Bressan. 2019. FACTCK.BR: A New Dataset to Study Fake News. In *Proceedings of the 25th Brazilian Symposium on Multimedia and the Web* (2019 ed.) (Rio de Janeiro, Brazil) (*WebMedia '19*, 1). Association for Computing Machinery, New York, NY, USA, 525–527. <https://doi.org/10.1145/3323503.3361698>
- [22] Mehwish Nasim, Andrew Nguyen, Nick Lothian, Robert Cope, and Lewis Mitchell. 2018. Real-Time Detection of Content Polluters in Partially Observable Twitter Networks. In *Companion Proceedings of the The Web Conference 2018* (2018 ed.) (Lyon, France) (WWW '18, .). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1331–1339. <https://doi.org/10.1145/3184558.3191574>
- [23] R. Pressman and B. Maxim. 2016. *Engenharia de Software - 8ª Edição*. <https://books.google.com.br/books?id=wexzCwAAQBAJ>
- [24] Feng Qian, Chengyue Gong, Karishma Sharma, and Yan Liu. 2018. Neural user response generator: fake news detection with collective user intelligence. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence* (Stockholm, Sweden) (IJCAI'18). AAAI Press, 3834–3840.
- [25] Adrian Rauchfleisch and Jonas Kaiser. 2020. In *The False positive problem of automatic bot detection in social science research*. Harvard Dataverse. <https://doi.org/10.7910/DVN/XVCKRS/P2ZKRU>
- [26] Natali Ruchansky, Sungyong Seo, and Yan Liu. 2017. CSI: A Hybrid Deep Model for Fake News Detection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (2017 ed.) (Singapore, Singapore) (CIKM '17, .). Association for Computing Machinery, New York, NY, USA, 797–806. <https://doi.org/10.1145/3132847.3132877>
- [27] Giovanni Santia and Jake Williams. 2018. BuzzFace: A News Veracity Dataset with Facebook User Commentary and Egos. *Proceedings of the International AAAI Conference on Web and Social Media* 12, 1 (Jun. 2018), 531–540. <https://doi.org/10.1609/icwsm.v12i1.14985>
- [28] Mohsen Sayyadifarikandeh, Onur Varol, Kai-Cheng Yang, Alessandro Flammini, and Filippo Menczer. 2020. Detection of Novel Social Bots by Ensembles of Specialized Classifiers. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* (Virtual Event, Ireland) (CIKM '20). Association for Computing Machinery, New York, NY, USA, 2725–2732. <https://doi.org/10.1145/3340531.3412698>
- [29] Chengcheng Shao, Giovanni Luca Ciampaglia, Onur Varol, Kai-Cheng Yang, Alessandro Flammini, and Filippo Menczer. 2018. The spread of low-credibility content by social bots. *Nature Commun.* 9, 1 (Nov. 2018). <https://doi.org/10.1038/s41467-018-06930-7>
- [30] Kai Shu, Deepak Mahadeswaran, and Huan Liu. 2019. FakeNewsTracker: a tool for fake news collection, detection, and visualization. *Computational and Mathematical Organization Theory* 25, 1 (March 2019), 60–71. <https://doi.org/10.1007/s10588-018-09280-1>
- [31] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake News Detection on Social Media: A Data Mining Perspective. *SIGKDD Explor. Newsl.* 19, 1 (sep 2017), 22–36. <https://doi.org/10.1145/3137597.3137600>
- [32] Paulo Márcio Souza Freire, Flávio Roberto Matias da Silva, and Ronaldo Ribeiro Goldschmidt. 2021. Fake news detection based on explicit and implicit signals of a hybrid crowd: An approach inspired in meta-learning. *Expert Systems with Applications* 183 (2021), 115414. <https://doi.org/10.1016/j.eswa.2021.115414>
- [33] Ajitesh Srivastava, Rajgopal Kannan, Charalampos Chelmis, and Viktor K. Prasanna. 2018. FActCheck: Keeping Activation of Fake News at Check. In *Adaptive Agents and Multi-Agent Systems, AAMAS*. <https://api.semanticscholar.org/CorpusID:51868670>
- [34] Duyu Tang, Furu Wei, Nan Yang, Ting Liu, and Ming Zhou. 2015. Sentiment Embeddings with Applications to Sentiment Analysis. *IEEE Transactions on Knowledge and Data Engineering* 28 (01 2015), 1–1. <https://doi.org/10.1109/TKDE.2015.2489653>
- [35] Patrick Wang, Rafael Angarita, and Ilaria Renna. 2018. Is This the Era of Misinformation yet: Combining Social Bots and Fake News to Deceive the Masses. In *Companion Proceedings of the The Web Conference 2018* (www '18 ed.) (Lyon, France). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1557–1561. <https://doi.org/10.1145/3184558.3191610>
- [36] Amy Warriner, Victor Kuperman, and Marc Brysbaert. 2013. Norms of valence, arousal, and dominance for 13,915 English lemmas. *Behavior research methods* 45 (02 2013). <https://doi.org/10.3758/s13428-012-0314-x>
- [37] Vinicius Woloszyn and Wolfgang Nejdl. 2018. DistrustRank: Spotting False News Domains. In *Proceedings of the 10th ACM Conference on Web Science* (Amsterdam, Netherlands) (*WebSci*). Association for Computing Machinery, New York, NY, USA, 221–228. <https://doi.org/10.1145/3201064.3201083>
- [38] Liang Wu and Huan Liu. 2018. Tracing Fake-News Footprints: Characterizing Social Media Messages by How They Propagate. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining* (Marina Del Rey, CA, USA) (WSDM '18). Association for Computing Machinery, New York, NY, USA, 637–645. <https://doi.org/10.1145/3159652.3159677>
- [39] Fan Yang, Shiva K. Pentylala, Sina Mohseni, Mengnan Du, Hao Yuan, Rhema Linder, Eric D. Ragan, Shuiwang Ji, and Xia (Ben) Hu. 2019. XFake: Explainable Fake News Detector with Visualizations. In *The World Wide Web Conference* (San Francisco, CA, USA) (WWW '19). Association for Computing Machinery, New York, NY, USA, 3600–3604. <https://doi.org/10.1145/3308558.3314119>
- [40] Kai-Cheng Yang, Onur Varol, Clayton A. Davis, Emilio Ferrara, Alessandro Flammini, and Filippo Menczer. 2019. Arming the public with artificial intelligence to counter social bots. *Human Behavior and Emerging Technologies* 1, 1 (Jan. 2019), 48–61. <https://doi.org/10.1002/hbe2.115>
- [41] Kai-Cheng Yang, Onur Varol, Pik-Mai Hui, and Filippo Menczer. 2020. Scalable and Generalizable Social Bot Detection through Data Selection. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 01 (April 2020), 1096–1103. <https://doi.org/10.1609/aaai.v34i0.15460>
- [42] Qiang Zhang, Eminie Yilmaz, and Shangsong Liang. 2018. Ranking-Based Method for News Stance Detection. In *Companion Proceedings of the The Web Conference 2018* (www '18 ed.) (Lyon, France). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 41–42. <https://doi.org/10.1145/3184558.3186919>