Detecção de *Fake News*: uma Abordagem Utilizando Redes Neurais Convolucionais^{*}

Matheus de Almeida Rosa^{*} Higor Coimbra Amorim^{*}
Rogério Martins Gomes^{*} Bruno Andre Santos^{*}

^{*} Departamento de Computação, Centro Federal de Educação
Tecnológica de Minas Gerais, Belo Horizonte, Minas Gerais, Brasil
(e-mail: mthalmeidarosa@gmail.com, higor1308@gmail.com,
rogerio@cefetmg.br, bsantos@cefetmg.br).

Abstract: Nowadays, the term *Fake News* has stood out all over the world. A good example of this was the Brazilian election in 2018, in which Fake News had a major impact on the election outcome. Considering these facts, it is appropriate to develop a mechanism, by using machine learning techniques, that can classify news as false or true. This paper proposes two methods of classification, convolutional neural networks (CNN) and artificial neural networks of multiple layers (ANN), in the detection of *Fake News*. These two methods were compared to two others found in the literature, using the same database. While in this work an accuracy of 96% for CNN and 95% for ANN was obtained, in the literature the highest accuracy was 89% for SVM (Support Vector Machine) and 90% for ANN.

Resumo: Hoje em dia, o termo *Fake News* tem se destacado em todo o mundo. Um bom exemplo disso, foi a eleição brasileira de 2018, na qual as *Fake News* tiveram um grande impacto no resultado das eleições. Considerando esses fatos, torna-se apropriado o desenvolvimento de um mecanismo, por meio das técnicas de aprendizado de máquina, que seja capaz de classificar notícias em falsas ou verdadeiras. Este artigo, propõe dois métodos de classificação, as redes neurais convolucionais (Convolutional Neural Network - CNN) e as redes neurais artificiais de múltiplas camadas (RNA), na detecção das *Fake News*. Estes dois métodos foram comparados a outros dois encontrados na literatura, utilizando a mesma base de dados. Enquanto neste trabalho obteve-se uma acurácia de 96% para a CNN e 95% para a RNA, na literatura as maiores acurácias foram de 89% para o SVN (Support Vector Machine) e 90% para a RNA.

Keywords: Fake News; Logistic Regression; Convolutional Neural Networks.

Palavras-chaves: *Fake News*; Regressão Logística; Redes Neurais Convolucionais.

1. INTRODUÇÃO

O número de usuários das redes sociais tem crescido de maneira vertiginosa devido a sua grande popularização nos últimos anos. Junto a isto, pode-se observar um aumento da sua influência como fonte de notícias, dado seu grande potencial de divulgação e a praticidade com que os usuários têm acesso a estas informações. No entanto, nos últimos anos, essas redes vêm sendo utilizadas para para influenciar a opinião pública, como por exemplo, nas eleições presidenciais, na qual procura-se, por meio da divulgação de notícias falsas, atacar a reputação dos candidatos e, dessa forma, mudar o resultado das urnas (Allcott and Gentzkow, 2017). Tais notícias se popularizaram como *Fake News*, e se tornaram um dos maiores problemas existentes nas redes sociais (Vosoughi et al., 2018; Lazer et al., 2018).

O termo *Fake News*, segundo Shu et al. (2017), consiste na distribuição deliberada de desinformação ou boatos via jornal impresso, televisão, rádio, ou, o mais comum,

por meio mídias sociais. O seu impacto é indiscutível no mundo atual e um bom exemplo disso são as eleições gerais brasileiras de 2018, cuja disseminação e influência das informações falsas chegaram em alguns momentos a ser maior que as verdadeiras. Apesar de popular hoje em dia, este assunto não é novo e, segundo Shu et al. (2017), desde a invenção da imprensa em 1439, já era possível encontrar relatos sobre a criação e veiculação de informações falsas em diferentes contextos e com diferentes objetivos.

Levando em consideração esses aspectos é fundamental a busca por formas de classificar uma notícia como verdadeira ou falsa dentro de um certo grau de confiança. A Inteligência Artificial, em especial o aprendizado de máquina, é uma ferramenta poderosa para realizar esta tarefa e, portanto, se encontra como uma das principais alternativas para a resolução do problema.

Na literatura é possível encontrar propostas semelhantes de detecção de *Fake News* utilizando *Deep Learning*. Singhanian et al. (2017), por exemplo, utilizou uma rede 3HAN, ou seja, uma rede hierárquica de atenção de 3 níveis (*3 levels hierarchical attention network*) que gera uma estrutura denominada *news vector* com capacidade

^{*} Os autores gostariam de agradecer ao CEFET-MG pelo suporte financeiro.

de classificar com uma determinada probabilidade, se uma notícia é *fake* ou não. Neste trabalho, os autores dividiram uma notícia em 3 níveis de acordo com a sua estrutura semântica e suas composições: palavras formam sentenças, sentenças formam o corpo e o corpo juntamente com a manchete formam o artigo. Assim, o artigo completo é então representado por interações entre os 3 níveis, levando em conta fatores como o contexto de uma palavra dentro do significado das suas palavras vizinhas, a coerência de uma sentença com as sentenças vizinhas e a relevância da manchete em relação ao corpo da notícia. As 3 estruturas de texto descritas (palavras, sentenças e manchete) são dependentes entre si na criação de um significado para a notícia. Sendo assim, o autor aproveitou-se desse fato para gerar uma estrutura classificadora, o *news vector*, com base na interação semântica das estruturas. Comparada a outras redes neurais, a rede 3HAN obteve uma performance superior, com uma acurácia de 96,24%. Quando um modelo pré-treinado, utilizando manchetes como forma de iniciar a rede, foi utilizado, a rede obteve uma acurácia de 96,77%.

O trabalho de Wang (2017) descreveu o desenvolvimento de uma base de dados de *Fake News*, contendo 12.836 declarações provenientes de um site de política americana, todas pré-classificadas em uma escala de totalmente falsas a totalmente verdadeiras. Estas declarações passaram por 6 graus de veracidade, além do desenvolvimento de uma arquitetura híbrida de redes neurais para melhorar a classificação de uma declaração como falsa. O autor definiu uma estrutura para o dado contendo, além do conteúdo de uma declaração, o seu contexto, o orador, a classificação de veracidade e uma justificativa da classificação. Além disso, os autores incluíram meta-dados sobre o orador, como sua afiliação partidária, trabalho atual e, o mais interessante deles, um histórico de declarações feitas e suas classificações de veracidade por parte do orador descrito.

Com o tamanho da base de dados desenvolvida, abriu-se a porta para inclusão de algoritmos de *Machine Learning* no aprendizado para classificação de veracidade das declarações. No trabalho descrito, foi construída uma arquitetura híbrida, envolvendo uma matriz de declarações, uma matriz de meta-dados dos oradores, uma camada de uma rede LSTM (*long short-term memory networks*) e uma rede neural convolucional (Convolutional Neural Network - CNN) para obtenção do melhor resultado em termos classificatórios. Os resultados obtidos por Wang (2017) mostraram que a arquitetura híbrida obteve melhor desempenho quando comparados à outros métodos de classificação tais como SVM (Support Vector Machine), regressões lineares, entre outros.

Gahirwal et al. (2018), por sua vez, desenvolveu uma arquitetura na qual um texto, após ser extraído de uma notícia, passou por pré-processamento responsável pela geração de uma estrutura de palavras a serem utilizadas nas próximas etapas da arquitetura. Neste trabalho, o principal fator para detecção de *Fake News* é a postura do autor quanto ao conteúdo da notícia. A postura foi dividida em Concordo, Discordo, Neutro ou Não relacionado, como forma de pesar se uma notícia é falsa ou não. O segundo fator para detecção de *fake News* foi a comparação da notícia avaliada com os melhores resultados em uma busca do Google, levando em consideração que tais resultados

Figura 1. Metodologia

são de extrema veracidade. A partir desses dois fatores, Gahirwal et al. (2018) utilizaram-se de algoritmos de classificação ou modelos de regressão para, após a classificação de uma notícia, avaliar pelo *F1-score* o teor de veracidade de uma notícia. Os autores do estudo não descreveram os testes realizados com essa arquitetura, não sendo possível, portanto, avaliar a eficiência dos métodos.

O presente artigo visa propor e discutir uma solução para o problema das *Fake News*, comparando 2 técnicas de aprendizado de máquina, ou seja, as **Redes Neurais Convolucionais (CNN)** e as **Redes Neurais Artificiais de Múltiplas Camadas (RNA)**.

2. METODOLOGIA

A metodologia usada neste trabalho consiste basicamente das seguintes etapas: escolha de uma base de dados de notícias em português do Brasil; mineração de dados; criação dos modelos de *machine learning* a serem aplicados na detecção de *fake news*; treinamento e análise de resultados. Todas essas etapas podem ser vistas na Figura 1.

A base de dados utilizada para treinamento dos modelos computacionais foi criada por pesquisadores da USP e é

chamada de *Fake.BR Corpus*. Esta base pode ser encontrada em: <https://github.com/roneysco/Fake.br-Corpus> e é composta por notícias verdadeiras e falsas, concentrando-se apenas no português do Brasil. Segundo Monteiro et al. (2018) não há outro corpus similar disponível nesta linguagem.

O Fake.BR Corpus possui dois diretórios principais, sendo um com notícias normalizadas (em tamanho) e outro com as notícias originais, ambos com 3.602 notícias falsas e 3.602 verdadeiras. Esta base possui também as meta-informações de cada uma das notícias, ou seja, nome, data, autor, entre outras. No entanto, por questões de simplificação, apenas o corpo das notícias será utilizado neste trabalho. A Tabela 1 mostra o número de documentos por categoria na base *corpus Fake.Br*, sendo possível observar a presença de um maior número de notícias relacionadas à política.

Tabela 1. Num. de documentos por categoria do *corpus Fake.Br* (Monteiro et al., 2018).

Categoria	Nº de amostras	%
Política	4.180	58%
TV e celebridade	1.544	21,4%
Sociedade	1.2761	17,7%
Ciência e tecnologia	112	1,5%
Economia	44	0,7%
Religião	44	0,7%

Neste artigo, todas as 7.204 notícias passam por um pré-processamento no qual caracteres estranhos e *noise corpus* são removidos. Estas etapas são descritas a seguir:

- Mineração de dados manual
 - *Remoção de caracteres estranhos*: devido ao fato das notícias serem reais, muitos caracteres se encontram em formato que não é reconhecido pelo processo de aprendizado de máquina. Sendo assim, estes caracteres foram removidos, de forma que todos os textos pudessem ser reconhecidos pelo padrão utf-8.
- Mineração de dados automática
 - *Remoção de "Noise Corpus"*: visando tornar os classificadores mais eficientes com relação ao domínio do problema, faz-se necessário a eliminação de partes do texto que possam não contribuir com o resultado. Ou seja, urls, *emoticons*, *hashtags*, números, símbolos e pontuação. Mais uma vez, em busca de uma maior simplicidade, foram mantidas apenas palavras na resolução do problema. No entanto, alguns corpos foram mantidos no problema de *Fake News*, como por exemplo, as *hashtags* ou datas. Esta eliminação foi feita utilizando-se expressões regulares.
 - *Remoção de "Stop Words"*: as *stop words* são palavras comumente utilizadas em qualquer texto e que conseqüentemente não são úteis na diferenciação dos mesmos (Uysal and Gunal, 2014). Portanto, palavras como "a", "uma", "cima", "diz", "haja", dentre outras, foram eliminadas dos textos. Esta remoção foi feita utilizando-se um método de força bruta, ou seja, cada palavra do texto foi comparada com uma lista prévia de *stop-words* e removida caso fosse igual ao elemento comparado. Neste passo, também foi realizada a

normalização do tamanho dos textos, reduzindo os mesmos para as 100 palavras iniciais. Além disso, todo o texto foi modificado para caixa baixa, padronizando, dessa forma, as palavras.

- *Stemming*: duas palavras podem possuir um prefixo comum, indicando, neste caso, uma intenção semelhante e que pode ser representada por tal prefixo (Uysal and Gunal, 2014). Portanto, visando simplificar o modelo e diminuir o número de variáveis, grupos de palavras, tais como "jogar", "jogo", "jogador", "joga", "joguei", "jogamos", podem ser representadas por um único símbolo "jog". O *stemming* das palavras foi realizado utilizando-se o *Natural Language Toolkit* do *Python*, que já possui uma base de dados com raízes de palavras em português.
- Frequência baixa: palavras que possuem menos de 10 ocorrências em todos os documentos foram removidas dado que este tipo de palavra raramente afetaria o desempenho da classificação.
- Mineração de dados do pré-modelo
 - Vetorização do texto: para que o modelo possa trabalhar numericamente com os dados, é necessário vetorizar o texto, isto é, transformar cada palavra em um número e cada texto em um vetor composto por tais números. Neste artigo, esse vetor é constituído de 44.205 posições que representam cada uma das possíveis palavras encontradas em um texto. Assim, cada posição do vetor mostra o número de ocorrências de cada uma dessas palavras. Caso a palavra não seja encontrada no texto, o valor da sua posição no vetor é ajustado para zero.
 - Incorporação de palavras: após transformar o texto em vetor, uma técnica bastante utilizada em NLP (*Natural Language Processing*) chamada de incorporação de palavras (*embedding words*) é utilizada (Zhou and Zhang, 2008) (Hauch et al., 2015). Esta técnica consiste em representar palavras com significados semelhantes de uma maneira semelhante. Para isso, as próprias palavras do texto são transformadas em vetores, utilizando-se mais uma vez uma ferramenta do Keras (Ketkar, 2017). É possível também fazer uso de um *pretrained embedding words* ou, adicionar o *embedding words* como mais uma camada a ser treinada da rede. Este trabalho utiliza a segunda opção.

Após a mineração, os dados estão prontos para serem utilizados, mas para isso é necessário a construção dos modelos. Este projeto utiliza a ferramenta Keras para construção dos modelos, que é uma ferramenta de código aberta escrita em cima de *Frameworks* robustos. Para este projeto foi utilizado o Keras executando sobre o *framework Tensor Flow*, do Google (Ketkar, 2017).

O primeiro modelo utilizado neste trabalho é uma rede neural convolucional, cujas camadas podem ser configuradas uma a uma. As camadas do modelo são descritas a seguir:

- *Embedding Layer*: Camada para treinar as semelhanças entre as palavras, criando um *embedding word*

data set. Esta é a primeira camada e, dessa forma, possui dimensão igual a do vocabulário total.

- Camada Convolutiva: esta camada convolutiva visa agrupar características da entrada de dados. Portanto, se inicialmente tem-se apenas palavras, após essa camada convolutiva, passa-se a ter grupos de palavras, ou seja, a saída passa a ser a identificação de “frases”. O número de filtros escolhido para a convolução foi 10 e a dimensão dos filtros utilizada foi igual a 5. Escolheu-se uma dimensão pequena para os filtros pois a separação de frases não possui uma grande quantidade de palavras.
- *Max Pooling*: como a saída da camada convolutiva possui diversos valores, é necessário definir um método de escolha de um deles como saída. O método utilizado no projeto foi o *Max Pooling*, que escolhe o maior dentre os valores da saída. Este método foi escolhido para dar ênfase a característica que mais se destacou no filtro.
- Planificação: Como a saída da convolução consiste em uma série de filtros é necessário transformar tal saída em uma entrada simples para a rede neural tradicional posterior a CNN. Para isso utilizou-se a camada *Flatten* do Keras.
- Camada relu: Uma camada simples de rede neural com saída relu foi utilizada para treinar utilizando as frases geradas pelas camadas convolutivas.
- Saída sigmoide: por ser um problema de classificação, a saída sigmoide atendeu o domínio do problema.

O segundo modelo utilizado neste artigo foi uma rede neural de múltiplas camadas. A arquitetura desta rede era constituída por uma camada de entrada de tamanho igual a 44.205, correspondente ao tamanho do vetor de palavras. Uma camada escondida, com função de ativação relu e com 25 neurônios, ajustados empiricamente. Uma camada de saída constituída por um neurônio com função de ativação sigmoide limiar de 0,5 para detecção de *fake news*.

O modelo foi configurado com uma função de custo de entropia cruzada (*binary cross-entropy*) e com o otimizador Adam, sendo a acurácia o parâmetro configurado para ser avaliado pelo modelo durante o treinamento e teste. Os resultados foram obtidos com a execução de 3 épocas.

Após todas as configurações, o *dataset* foi dividido em dois conjuntos disjuntos de dados, na proporção de 2/3 para treino e 1/3 para teste, e foram aplicados nos dois modelos propostos neste artigo. Os resultados dos experimentos podem ser vistos na próxima seção.

2.1 Resultados

O resultado obtido para a arquitetura desenvolvida utilizando a CNN pode ser visto na Tabela 2. Na Tabela 3, por sua vez, é mostrada a matriz de confusão deste experimento.

Tabela 2. Acurácia, Precisão, *Recall* e *F1 score* para a CNN

Acurácia	0,9630
Precisão	0,9576
Recall	0,9667
F1 score	0,9621

Tabela 3. Matriz de confusão para a CNN

	true:fake	true:real
pred:fake	1016	45
pred:real	35	1064

A seguir, o resultado do experimento com as Redes Neurais Convolucionais foi comparado com o experimento envolvendo as Redes Neurais Artificiais de Múltiplas Camadas otimizada. Os resultados obtidos de Acurácia, Precisão, *Recall* e *F1 Score* para a RNA otimizada são mostrados na Tabela 4, enquanto a matriz de confusão pode ser vista na Tabela 5. Pode-se observar que os resultados obtidos nos dois métodos foram, de uma maneira geral, muito próximos. No entanto, a CNN, obteve um *F1 score* superior, apresentando, dessa forma, uma melhor média harmônica entre a precisão e o *recall*.

Outro aspecto importante na análise do desempenho dos modelos de predição deve ser considerado, ou seja, errar a classificação de uma notícia verdadeira é mais prejudicial do que não detectar algumas notícias falsas. Sendo assim, para bases de notícias desbalanceadas seria melhor, dentro deste contexto e para um mesmo valor de *F1 score*, que os métodos apresentassem um valor mais elevado de precisão do que de *recall*.

Tabela 4. Acurácia, Precisão, *Recall* e *F1 score* para a RNA

Acurácia	0,9523
Precisão	0,9608
Recall	0,9418
F1 score	0,9512

Tabela 5. Matriz de confusão para a RNA

	true:fake	true:real
pred:fake	1004	41
pred:real	62	1053

Após a análise dos dois métodos propostos neste artigo, foi realizada uma comparação com os resultados obtidos em (Monteiro et al., 2018), que utilizou a mesma base de dados. No trabalho de Monteiro et al. (2018) o melhor resultado obtido foi de 90% de acurácia, quando os autores utilizaram redes neurais de múltiplas camadas e os parâmetros padrão do pacote *Scikit-learn*. Outro método desenvolvido pelos autores foi o SVM, também com os parâmetros padrão do pacote *Scikit-learn*, quando os autores alcançaram uma acurácia máxima de 89%. Sendo assim, pode-se perceber que os resultados obtidos neste trabalho foram superiores aos encontrados em (Monteiro et al., 2018).

3. CONCLUSÕES

Este artigo se propôs a realizar uma comparação entre três métodos de *machine learning*: as redes neurais convolucionais (CNN), as redes neurais artificiais de múltiplas camadas otimizada e as redes neurais com os parâmetros padrão do pacote *Scikit-learn* proposta em (Monteiro et al., 2018). Para isso, realizou uma análise quantitativa da acurácia, precisão, *recall*, *F1 score*, bem como da matriz de confusão. Ao final, foi possível observar que, tanto as CNN quanto as redes neurais artificiais múltiplas camadas otimizada,

apresentaram resultados superiores aos encontrados na literatura para a mesma base de dados. Sendo assim, este trabalho abre um leque de possibilidades para o uso de métodos de *machine learning* na detecção de *Fake News*.

Como trabalho futuro, propõe-se a análise da arquitetura CNN, proposta neste trabalho, para diversas base de dados de notícias, gerando resultados mais generalizados ao problema de detecção de *Fake News*.

AGRADECIMENTOS

Os autores gostariam de agradecer ao Laboratório de Sistemas Inteligentes (LSI) pela infraestrutura disponibilizada e ao CEFET-MG pelo suporte financeiro, sem o qual esse trabalho não teria sido viável.

REFERÊNCIAS

- Allcott, H. and Gentzkow, M. (2017). Social media and fake news in the 2016 election. *Journal of economic perspectives*, 31(2), 211–36.
- Gahirwal, M., Moghe, S., Kulkarni, T., Khakhar, D., and Bhatia, J. (2018). Fake news detection. *International Journal of Advance Research, Ideas and Innovations in Technology*, 4(1), 817–819.
- Hauch, V., Blandón-Gitlin, I., Masip, J., and Sporer, S.L. (2015). Are computers effective lie detectors? a meta-analysis of linguistic cues to deception. *Personality and Social Psychology Review*, 19(4), 307–342.
- Ketkar, N. (2017). Introduction to keras. In *Deep Learning with Python*, 97–111. Springer.
- Lazer, D.M., Baum, M.A., Benkler, Y., Berinsky, A.J., Greenhill, K.M., Menczer, F., Metzger, M.J., Nyhan, B., Pennycook, G., Rothschild, D., et al. (2018). The science of fake news. *Science*, 359(6380), 1094–1096.
- Monteiro, R.A., Santos, R.L., Pardo, T.A., de Almeida, T.A., Ruiz, E.E., and Vale, O.A. (2018). Contributions to the study of fake news in portuguese: New corpus and automatic detection results. In *International Conference on Computational Processing of the Portuguese Language*, 324–334. Springer.
- Shu, K., Sliva, A., Wang, S., Tang, J., and Liu, H. (2017). Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, 19(1), 22–36.
- Singhania, S., Fernandez, N., and Rao, S. (2017). 3han: A deep neural network for fake news detection. In *International Conference on Neural Information Processing*, 572–581. Springer.
- Uysal, A.K. and Gunal, S. (2014). The impact of preprocessing on text classification. *Information Processing & Management*, 50(1), 104–112.
- Vosoughi, S., Roy, D., and Aral, S. (2018). The spread of true and false news online. *Science*, 359(6380), 1146–1151.
- Wang, W.Y. (2017). ”liar, liar pants on fire”: A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648*.
- Zhou, L. and Zhang, D. (2008). Following linguistic footprints: automatic deception detection in online communication. *Commun. ACM*, 51(9), 119–122.