UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

UMA PROPOSTA DE ARQUITETURA DE BIG DATA PARA DETECÇÃO DE FAKE NEWS

DANIELE MOURA DE QUEIROZ

DM 03/2020

UFPA/ITEC/PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil
2020

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

DANIELE MOURA DE QUEIROZ

UMA PROPOSTA DE ARQUITETURA DE BIG DATA PARA DETECÇÃO DE FAKE NEWS

DM 03/2020

UFPA/ITEC/PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil
2020

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

DANIELE MOURA DE QUEIROZ

UMA PROPOSTA DE ARQUITETURA DE BIG DATA PARA DETECÇÃO DE FAKE NEWS

Dissertação submetida à Banca Examinadora do Programa de Pós-Graduação em Engenharia Elétrica da UFPA para a obtenção do Grau de Mestre em Engenharia Elétrica com ênfase em Computação Aplicada sob a orientação do Prof. Dr. Carlos Renato Lisboa Francês.

UFPA/ITEC/PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil
2020

Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD
Sistema de Bibliotecas da Universidade Federal do Pará
Gerada automaticamente pelo módulo Ficat, mediante os dados fornecidos pelo(a) autor(a)

Q3p Queiroz, Daniele Moura de
UMA PROPOSTA DE ARQUITETURA DE BIG DATA PARA DETECÇÃO DE FAKE NEWS / Daniele Moura de Queiroz. — 2020.
72 f. : il. color.

Orientador(a): Prof. Dr. Carlos Renato Lisboa Francês
Dissertação (Mestrado) - Programa de Pós-Graduação em Engenharia Elétrica, Instituto de Tecnologia, Universidade Federal do Pará, Belém, 2020.

1. Big data. 2. Fake news. 3. Arquitetura de big data. 4. Hadoop. I. Título.

CDD 004

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

"UMA PROPOSTA DE ARQUITETURA DE BIG DATA PARA DETECÇÃO DE FAKE NEWS"

AUTORA: DANIELE MOURA DE QUEIROZ

DISSERTAÇÃO DE MESTRADO SUBMETIDA À BANCA EXAMINADORA APROVADA PELO COLEGIADO DO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA, SENDO JULGADA ADEQUADA PARA A OBTENÇÃO DO GRAU DE MESTRA EM ENGENHARIA ELÉTRICA NA ÁREA DE COMPUTAÇÃO APLICADA.

APROVADA EM: 24/01/2020

BANCA EXAMINADORA:

Prof. Dr. Carlos Renato Lisboa Francês
(Orientador – PPGEE/UFPA)

Prof.ª Dr.ª Jasmine Priscyla leite de Araújo
(Avaliadora Interna – PPGEE/UFPA)

Prof.ª Dr.ª Silvana Rossy de Brito
(Avaliadora Externa – UFRA)

VISTO:

Prof.ª Dr.ª Maria Emília de Lima Tostes
(Coordenadora do PPGEE/ITEC/UFPA)

Dedico este trabalho a Deus, à minha família e aos amigos que me acompanharam nesta jornada.

AGRADECIMENTOS

A Deus pelo dom da vida, por sua misericórdia e por estar comigo todos os dias e em todos os momentos, iluminando meus caminhos e me protegendo.

Aos meus pais, José Elidio Queiroz e Leuzi Queiroz, por todo o amor, carinho e paciência que dedicaram durante todas as fases da minha vida e especialmente pelo incentivo aos estudos, o que nos proporcionou mudanças profundas e maravilhosas em nossas vidas.

À minha irmã Gisele Queiroz, minha melhor amiga, pelo incentivo em todas as minhas lutas, pelos conselhos sinceros, por sofrer junto comigo nas minhas batalhas e comemorar minhas conquistas.

Ao meu namorado e parceiro de todas as horas, Vinícius Freitas, que compartilha e desfruta a vida comigo e que tanto me incentivou e contribuiu para que eu persistisse e concluísse mais esta etapa da minha vida.

Ao meu orientador Prof. Dr. Carlos Renato Lisboa Francês, pela paciência, confiança, orientação, competência, dedicação e oportunidade de realizar este trabalho. Obrigada por ter me acolhido, por me tratar com carinho e por sua amizade.

As amigas Kelle Costa, Lena Veiga e Maria da Penha que tanto me apoiaram durante esta jornada, através de incentivos, ideias, dicas e companheirismo, tornando essa caminhada mais leve e prazerosa.

Aos colegas do Laboratório de Planejamento de Redes de Alto Desempenho – LPRAD e Laboratório de Tecnologias Sociais – LTS da Universidade Federal do Pará, que de alguma maneira contribuíram para este trabalho, em especial: Hugo Kuribayashi, Aurea Santos, Rodrigo Alfaia, Anderson Souto, Jonatã Paulino e Priscila Aranha.

Aos professores e funcionários do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Pará pelo apoio e aprendizado, em especial a Profa. Dra. Jasmine Araújo e ao Prof. Dr. Marcelino Silva, por terem contribuição especial ao meu aprendizado.

“Até aqui nos ajudou o Senhor.”

(I Samuel 7:12)

RESUMO

Nos últimos anos, uma grande quantidade de informações tem sido veiculada através da internet, especialmente em mídias sociais, proporcionando uma maior facilidade na obtenção de conhecimentos sobre diversos temas, mas tornando as pessoas suscetíveis a informações falsas que podem acarretar danos variados. Apesar de não ser um fenômeno recente, o compartilhamento de notícias falsas tem sido motivo de preocupação por parte de especialistas e da população em geral, uma vez que pode ocasionar impactos de proporções nacionais e até mundiais. A veiculação de fake news pode ocasionar danos diversos, desde financeiros a prejuízos relacionados com difamação, injúria, ofensa, reputação ou dignidade de pessoas ou organizações. A disseminação destas informações falsas tornou difícil detectar fontes de notícias confiáveis, aumentando a necessidade de ferramentas computacionais que sejam capazes de auxiliar na identificação da confiabilidade do conteúdo digital. Ademais, a quantidade massiva de dados gerados diariamente em alta velocidade e com diferentes tipos de formatos tais como textos, imagens, vídeos e áudios, torna a análise destes dados um grande desafio. Com o advento das tecnologias de Big Data, é possível utilizar uma gama de ferramentas e técnicas para armazenar, processar e analisar, de maneira eficiente, o massivo volume de dados, de forma a contribuir com a investigação da credibilidade de notícias divulgadas e compartilhadas por meio da internet. Este trabalho discute a importância do Big Data para o combate às fake news, pautado em um apropriado enquadramento conceitual e tecnológico, e apresenta uma proposta de arquitetura de Big Data para armazenamento, processamento e análise de grandes conjuntos de dados, objetivando auxiliar na investigação da veracidade de notícias. Para tanto, foram realizados experimentos utilizando uma massa de dados contendo formatos distintos, ou seja, dados estruturados e não estruturados, extraídos a partir de fontes de notícias e formando um corpus composto por notícias falsas e verdadeiras, totalizando 10700 notícias. Essa massa de dados foi armazenada em um cluster Hadoop, utilizando o sistema de arquivos distribuído denominado HDFS (Hadoop Distributed File System). O processamento do corpus ocorreu através do modelo de programação MapReduce e a classificação das notícias foi realizada através do algoritmo Naive Bayes da biblioteca Mahout, obtendo-se uma acurácia de 99,74%. Os resultados preliminares produzidos pelo desenvolvimento deste estudo revelam uma arquitetura capaz de armazenar, processar e analisar Big Data no contexto do combate às fake news.

Palavras chaves: *Big Data; Fake News; Arquitetura de Big Data; Hadoop.*

ABSTRACT

In last years, a large amount of information has been transmitted through the Internet, especially in social media, providing greater knowledge on various topics, but making people susceptible to false information that can cause various damage. Although it is not a recent phenomenon, the sharing of false news has been a matter of concern for specialists and the population in general, since it can cause impacts of national and even global proportions. The transmission of fake news can cause various damages, from financial to losses related to defamation, injury, offense, reputation or dignity of people or organizations. The spread of this false information has made it difficult to detect reliable news sources, increasing the need for computational tools that can help identify the reliability of digital content. Moreover, the massive amount of data generated daily at high speed and different types of formats such as text, images, videos and audios, makes analysing this data a big challenge. With the advent of big data technologies, it is possible to use a range of tools and techniques to efficiently store, process and analyse massive data to help investigate the credibility of news disseminated and shared by middle of the internet. This paper discusses the importance of Big Data to combat fake news, based on an appropriate conceptual and technological framework, and presents a Big Data architecture proposal for storing, processing and analysing large data sets, aiming to assist in the investigation of truth of news. For this, experiments were performed using a mass of data containing different formats, i.e. structured and unstructured data, extracted from news sources and forming a corpus composed of false and true news. This mass of data was stored in a Hadoop cluster using the Hadoop Distributed File System (HDFS). The corpus was processed using the MapReduce programming model and the news classification was performed using the Naive Bayes algorithm from Mahout library, obtaining an accuracy of 99.74%. The preliminary results produced by the development of this study reveal an architecture capable of storing, processing and analysing Big Data in the context of fighting fake news.

Keywords: *Big Data; Fake News; Big Data Architecture; Hadoop.*

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS

PIB	Produto Interno Bruto
SMP	Symmetric Multi Processing
MPP	Massive Parallel Processing
NFS	Network File System
GFS	Google File System
HDFS	Hadoop Distributed File System
ADMM	Alternating Direction Method of Multipliers
KDD	Knowledge Discovery in Databases
NLP	Natural Language Processing
API	Application Programming Interface
JDBC	Java Database Connectivity
IA	Inteligência Artificial

LISTA DE FIGURAS

LISTA DE GRÁFICOS

LISTA DE TABELAS

1. INTRODUÇÃO

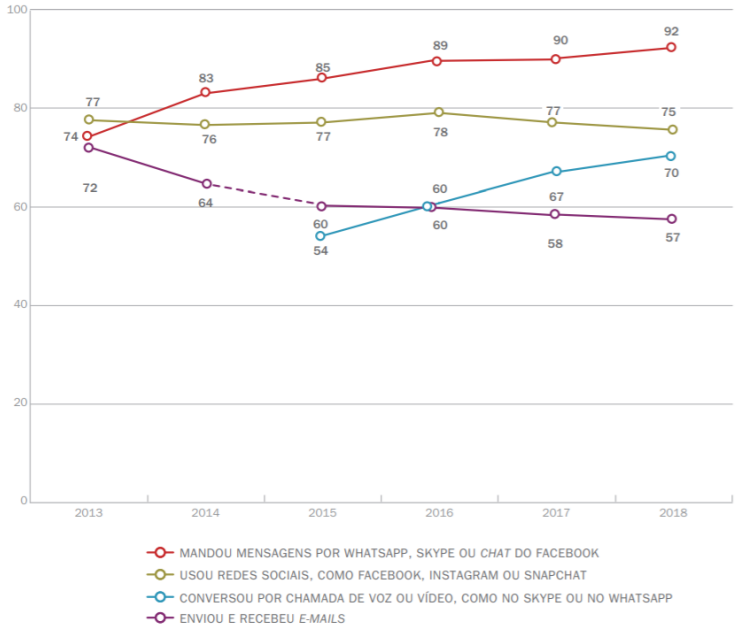
1.1. Motivação e Caracterização do Problema

A popularização do uso de computadores favorece o surgimento de novas demandas tanto no meio corporativo como no cotidiano das pessoas. Em decorrência disso, o acesso ao mundo digital passou a ser um novo indicador social que, capaz de definir os limites entre a igualdade de oportunidades e a exclusão social, acaba sendo um dos fatores que move a máquina pública a desenvolver estratégias de mercado que promovam a pesquisa, desenvolvimento e implantação de meios de comunicação e infraestrutura de tecnologia.

A exemplo disso pode-se citar o programa de inclusão digital intitulado NavegaPará, que oferece banda larga de acesso gratuito a agentes públicos, como municípios, escolas, hospitais, delegacias e prédios de escritórios (De Souza et al., 2008). Segundo SectetPA (2019), o NavegaPará consolida-se, hoje, como uma das maiores iniciativas públicas voltadas à democratização dos recursos da informática e da *internet* no Brasil. O projeto uniu forças com a Rede Metrobel, projeto de rede metropolitana que interliga instituições científicas da região metropolitana de Belém com fibra óptica própria e, com a Eletronorte, por meio do qual a estatal cedeu, ao estado do Pará, os 1.800 km de sua rede de fibra óptica (Castro; Baía, 2012).

Manobras políticas dessa natureza resultam em dados que o Comitê Gestor de Internet no Brasil (2019) ilustra em sua pesquisa que vem sendo realizada sistematicamente desde 2005. O estudo revela que em 2018 70% dos brasileiros usou a *internet* nos três meses que antecederam o estudo, o que corresponde a 126,9 milhões de pessoas. No recorte por classe socioeconômica, houve avanço no percentual de usuários das classes DE, passando de 30% em 2015 para cerca de metade da população em 2018 (48%). Confirmando a tendência de anos anteriores, os dispositivos móveis representam 97% do acesso à rede pelos brasileiros e, dentre as atividades *online* mais recorrentes estão: o envio de mensagens instantâneas, representando 92% e o uso de redes sociais, com 75%, conforme ilustra o Gráfico 1.

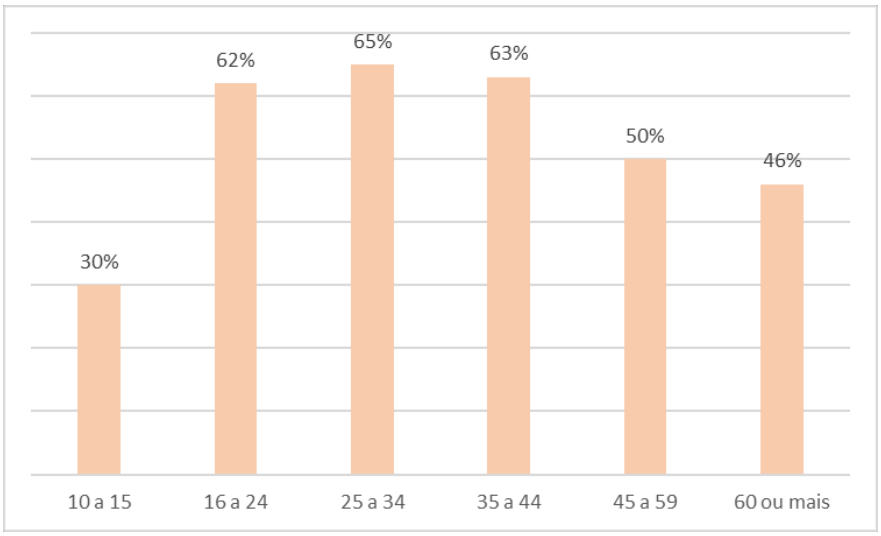
Gráfico 1: Usuários de *Internet*, por Atividades realizadas na *Internet*.



Fonte: Comitê Gestor de Internet no Brasil, 2019.

Os dados do TIC Domicílios 2018 mostram que as práticas mais comuns entre os usuários variam de acordo com a idade. Entre o público jovem (de 10 a 15 anos), a preferência é pela prática de jogos, por outro lado, quanto mais idade tiver o usuário maior é o interesse pela prática de leitura de jornais, especialmente entre os indivíduos de 25 a 34 anos. O Gráfico 2 representa o percentual de usuários de *internet* que leram jornais, revistas ou notícias *online*, de acordo com a faixa etária.

Gráfico 2: Usuários de Internet que Leram Notícias.



Fonte: Adaptado de Comitê Gestor de Internet no Brasil, 2019.

Diversos estudos comprovaram a importância do acesso à *internet* para o desenvolvimento econômico. Em Qiang, Rossotto e Kimura (2009) foram analisados 120 países em desenvolvimento e concluiu-se que o aumento de 10% no acesso à *internet* implica em 1,38% no crescimento do Produto Interno Bruto (PIB) *per capita*. Macedo e Carvalho (2010) também encontraram um relacionamento positivo entre o aumento da penetração do serviço de acesso à *internet* em banda larga e indicadores econômicos relativos ao PIB e ao PIB *per capita*. Além do crescimento econômico, Deloitte (2014) apresenta outros benefícios gerados a partir do aumento da conectividade, dentre os quais destacam-se: melhorias nas condições de saúde, reduzindo a incidência de doenças através do acesso a informações para pacientes e profissionais, perpassando ainda pela prevenção, diagnóstico, tratamentos e monitoramentos, a partir do contato entre paciente e médico ou entre médicos em diferentes locais, com impacto particularmente significativo nas comunidades rurais; aumento no acesso à educação, através de diversos recursos de aprendizado, permitindo, por exemplo, que um estudante em um país em desenvolvimento possa acessar a biblioteca de uma universidade de prestígio em qualquer lugar do mundo ou que professores possam se atualizar ou obter inspiração e conselhos através dos recursos e experiências de outras pessoas, ou seja, que o conhecimento seja amplamente disseminado; melhoria na eficiência dos serviços públicos; inclusão digital e desenvolvimento social.

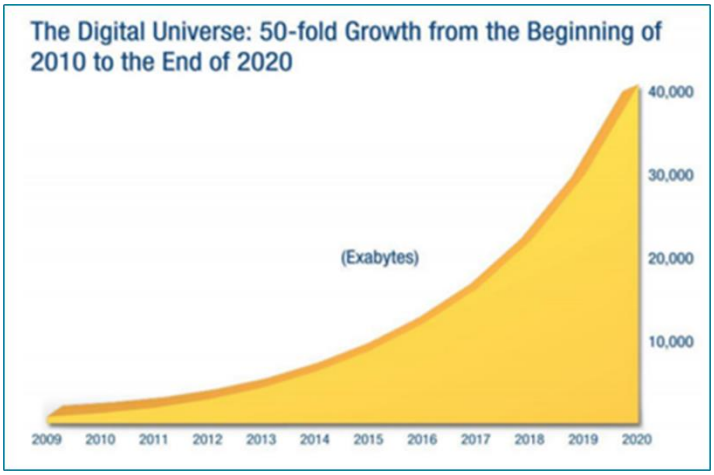
Ainda no que tange os impactos positivos, a *internet* tem viabilizado diversos protestos ao redor do mundo em países onde existe algum grau de democracia, permitindo a articulação desses protestos através de comunicações via *internet*. Também tem construído e/ou fortalecido relações humanas, como relacionamentos amorosos, empregos, amizades, etc.

Nota-se, portanto, uma acelerada democratização e popularização de acesso ao mundo digital que, aliada a necessidade de controle de informações do mundo moderno, move a proliferação de serviços na *internet* e o uso massivo de tecnologias da informação que têm sido responsáveis pelo crescimento exponencial da esmagadora quantidade de dados gerados e disponibilizados atualmente.

No âmbito dessa questão, Gantz e Reinsel (2012) ilustra que o volume de dados gerados mundialmente deu um salto de 166 *Exabytes* para 988 *Exabytes* no período de

2006 a 2010. Sua estimativa é que esse volume cresça cerca de 40 vezes até 2020, atingindo a casa dos 40000 *Exabytes*, conforme apresentado no Gráfico 3.

Gráfico 3: Perspectiva de Crescimento do Volume de Dados Digitais entre 2010 e 2020.



Fonte: Gantz e Reinsel, 2012.

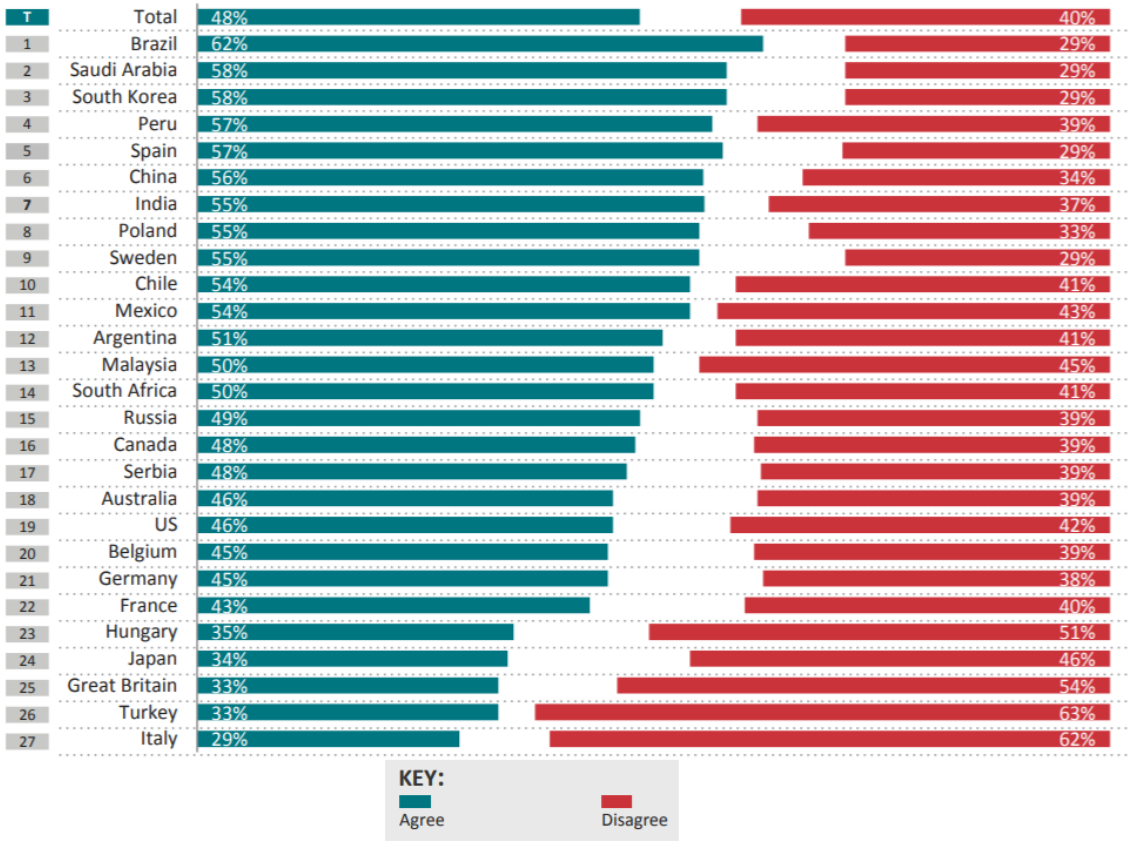
Apesar de resultar em grandes e profundas mudanças na sociedade, em sua maioria positivas, a *internet* pode ser palco de investidas criminosas. Uma das mais discutidas na atualidade é denominada *fake news*. O conceito de *fake news* tem atraído diversos campos de pesquisa em busca de uma delimitação do fenômeno, porém não há um consenso na definição do termo. Segundo Carillet (2019), o termo *fake news* é usado para referir-se a falsas informações divulgadas, principalmente, em redes sociais. Burshtein (2017) define a expressão como um relato fictício relativo aos eventos atuais que são fabricados e muitas vezes intitulados de forma enganosa, com o propósito deliberado de enganar os usuários e motivá-los a divulgar. Completa ainda que o significado do termo muitas vezes é mal utilizado para referir-se a notícias erradas, propagandas, sátiras e até fatos com os quais alguém não concorda. Allcott e Gentzkow (2017) afirmam tratar-se de artigos noticiosos que são intencionalmente falsos e aptos a serem verificados como tal, e que podem enganar os leitores. No jornalismo, autores defendem que *fake news* não remete a um objeto de seus estudos, já que se é notícia, não pode ser falsa e, se é falsa, não pode ser notícia (Braga, 2018).

A propagação de notícias falsas não é algo recente, porém tem se popularizado cada vez mais com o advento das mídias sociais, intensificando o potencial de persuasão que o material falso tem adquirido nos últimos anos. O termo começou a ser utilizado com mais frequência pela imprensa internacional durante as eleições presidenciais dos Estados Unidos em 2016, quando houve muitos debates pautados em vazamentos de

informações dos candidatos (Carillet, 2019). As notícias falsas, em grande maioria, são criadas e divulgadas com o intuito de legitimar um ponto de vista, prejudicar uma pessoa ou grupo, obter vantagens comerciais ou disseminar o ódio. Apesar de figuras públicas serem alvos constantes deste fenômeno, pessoas comuns também sofrem os efeitos dessa onda de mentiras e difamações.

Devido a crescente utilização de mídias sociais e busca de informações através da *internet*, notícias diversas tendem a ser acessadas e disseminadas a partir de fontes não oficiais. As razões para isso são diversas, e envolvem: (i) muitas vezes, é mais oportuno e mais barato consumir notícias nas redes sociais em comparação com a mídia tradicional, como jornais ou televisão; e (ii) é mais fácil compartilhar, comentar e discutir as notícias com amigos ou outros leitores nas redes sociais (Shu, 2017). Além disso, segundo o psiquiatra Claudio Martins, quando a pessoa recebe uma notícia que a agrada, são estimulados os mecanismos de recompensa imediata do cérebro e dão uma sensação de prazer instantâneo, assim como as drogas. Ocorre uma descarga emocional e gera uma satisfação imediata. Isso impulsiona a pessoa a transmitir compulsivamente a mesma informação para que seu círculo de amizades sinta o mesmo (Souza, 2018). Ainda segundo o psiquiatra, as informações não checadas são então transmitidas e são capazes de gerar uma curiosidade ampliada em outras pessoas, além de um alto nível de identificação e propagação de conteúdo, sendo o campo da política um dos mais propícios para esse fenômeno (Souza, 2018). O impacto das *fake news* pode ser observado na pesquisa de Duffy (2018), a qual aponta que dentre 27 países, o Brasil é o país com o maior número de pessoas que já acreditaram em uma notícia que, na verdade, era boato, conforme mostra o Gráfico 4. A maioria apontou acreditar nas informações que recebem, a não ser que exista uma indicação de irregularidade.

Gráfico 4: Acreditar Falsamente que uma Notícia era Real.



Fonte: Duffy, 2018.

À luz do impressionante volume de dados digitais gerados e compartilhados, detectar notícias falsas tem sido desafiador em diversos aspectos, que podem ser mitigados pela adoção de uma arquitetura de armazenamento e processamento de *big data* adequada. Para tal, torna-se fundamental o enquadramento conceitual e tecnológico da temática aqui investigada, incluindo a análise de iniciativas já existentes, de modo a identificar as suas características e limitações, permitindo, assim, evoluir para a especificação de uma arquitetura de *big data* no contexto de *fake news*.

1.2. Objetivo Geral

O objetivo geral desta dissertação é propor uma arquitetura de *big data* para auxiliar na detecção de notícias falsas, apresentando os resultados de experimento desenvolvido a partir da arquitetura concebida para armazenar e processar um conjunto heterogêneo de notícias, sobre os quais aplica-se técnica de aprendizado de máquina para classificar notícias e, assim, permitir a detecção de *fake news*.

1.3. Objetivos Específicos

A realização do objetivo geral está associada com os objetivos específicos enumerados a seguir:

- Realizar um amplo levantamento do estado da arte das áreas contempladas nesta dissertação;
- Descrever as ferramentas e técnicas mais utilizadas para armazenamento, processamento e análise de *big data*;
- Instalar e configurar as ferramentas da arquitetura de *big data* proposta;
- Realizar a coleta, o armazenamento e o tratamento de um conjunto heterogêneo de notícias na infraestrutura criada, compondo uma base de dados de notícias;
- Aplicar à base de dados técnica de aprendizado de máquina, buscando obter uma classificação das notícias em dois conjuntos: verdadeiras e falsas, a fim de validar a solução proposta.

1.4. Organização do Documento

Este trabalho está organizado da seguinte forma:

- O Capítulo 2 apresenta a fundamentação teórica relativa ao *big data*, incluindo ferramentas e técnicas aplicadas.
- No Capítulo 3 são apresentados os trabalhos relacionados ao estudo realizado.
- O Capitulo 4 apresenta a arquitetura proposta, através da especificação de várias camadas de abstração e dos constituintes de cada camada, quer em termos conceituais, quer em termos tecnológicos.
- No Capítulo 5 evidenciam-se os resultados de experimentação para validar a arquitetura proposta.
- E no Capítulo 6 são discutidas as conclusões, contribuições, dificuldades e propostas de trabalhos futuros.

2. REFERENCIAL TEÓRICO

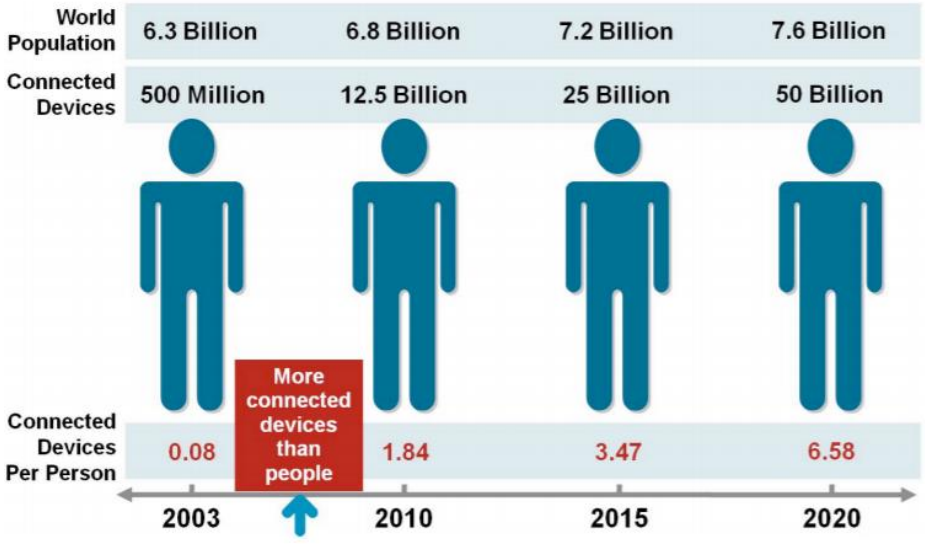
2.1. Considerações Iniciais

Este capítulo tem por finalidade conceituar teoricamente o conhecimento referente ao que foi aplicado neste trabalho. São abordados os principais conceitos necessários para a realização desta dissertação, assim como a análise de ferramentas que possam auxiliar na implementação da solução de *big data* proposta.

2.2. Big Data

De acordo com Bryant, Katz e Lazowska (2008), os avanços em sensores digitais, comunicações, computação e armazenamento criaram enormes coleções de dados, capturando informações de valor para negócios, ciência, governo e sociedade. O crescimento expressivo do número de dispositivos conectados à *internet* no mundo, em comparação ao aumento da população humana, vem resultando na geração de um intenso volume diário de dados. Em 2010 o número de dispositivos conectados superou o número da população mundial, mais precisamente 1,84 dispositivos conectados para cada pessoa, e, estima-se que em 2020, esse número terá uma proporção de mais de 6,58 vezes o número de habitantes no mundo (Evans, 2011), conforme ilustrado na Figura 1. Neste contexto emerge o fenômeno denominado *big data*.

Figura 1: Quantidade de Pessoas x Dispositivos Conectados.



Fonte: Evans, 2011.

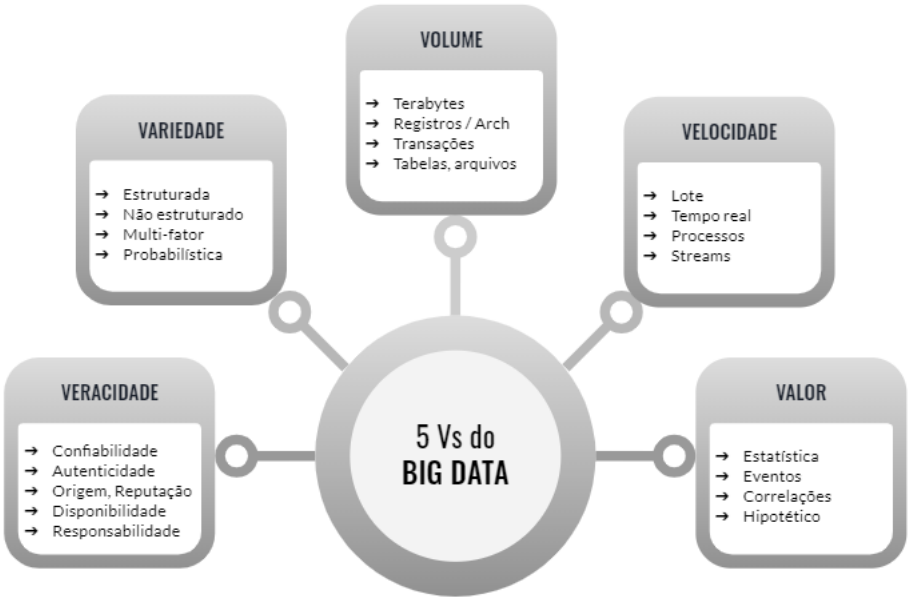
O termo *big data* é bastante amplo e não existe um consenso comum em relação à sua definição. Segundo Schutt e O'Neil (2013), *Big Data* pode ser resumidamente definido como o processamento (eficiente e escalável) analítico de um grande volume de dados complexos, que, em aplicações tradicionais, o seu processamento seria incomportável. De acordo com Gantz e Reinsel (2011), *Big Data* consiste em uma nova geração de tecnologias e arquiteturas projetadas para extrair economicamente valor de volumes muito grandes de uma ampla variedade de dados, permitindo captura, descoberta e/ou análise em alta velocidade. E segundo o Gartner (2019), *Big Data* é determinado por ativos de informações de alto volume, alta velocidade e/ou alta variedade que exigem formas inovadoras e econômicas de processamento de informações que permitam uma visão aprimorada, tomada de decisão e automação de processos.

Embora normalmente *big data* esteja associado a grandes volumes de dados, sua definição é dada por um conjunto de três a cinco "Vs". Inicialmente, a definição para "Vs" é de dados produzidos com volume, velocidade e variedade. Para dois "Vs" a mais, aparecem outras definições: veracidade e valor (Amaral, 2016). O modelo dos 3 "Vs" foi apresentado em 2001 por Douglas Laney, então analista da *Gartner*, modelo este usado pela *IBM* e *Microsoft* durante, pelo menos, os 10 anos seguintes (Chen et al., 2014). Porém, além da natureza volumosa, variada e veloz dos dados, temos que compreender o potencial da informação que pode ser extraída e analisada para revelar novo conhecimento e otimizar o processo de decisão (Sagiroglu; Sinanc, 2013). Uma perspectiva realista e prática acerca do modelo dos 3 "Vs" (volume, variedade e velocidade), é apresentada por Krishnan (2013), o qual afirma que o volume dos dados caracteriza a quantidade de dados gerados continuamente e diferentes tipos de dados possuem diferentes tamanhos. Isto remete para a variedade dos dados, que denota os múltiplos formatos possíveis: estruturados, semiestruturados ou não estruturados, provenientes de páginas *web*, arquivos de *log*, pesquisas, redes sociais, *e-mail*, documentos, dados de sensores, entre outros. Por fim, a velocidade consiste no fato de os dados fluírem de forma contínua, com o objetivo de serem adquiridos e processados num espaço de tempo muito curto, para retornarem um conjunto de resultados esperados. Zikopoulos e Eaton (2011) indicam que a principal causa para o aumento progressivo do volume de dados é o fato de armazenarmos todos os eventos que ocorrem nas nossas interações com a maioria dos serviços existentes no nosso mundo. O crescimento da quantidade de sensores, *smart devices* e o uso de diversas tecnologias contribui para a

variedade dos formatos de dados, cujo armazenamento, muitas das vezes, não é adequado em tecnologias de bases de dados tradicionais.

Com o crescente interesse da comunidade científica e técnica no conceito de *big data*, autores estudaram e acrescentaram novas características ao modelo dos 3 "Vs". Uma delas, denominada valor, revela a importância de armazenar, processar e analisar *big data*. O valor dos dados pode ser entendido como a importância da informação escondida no *big data*, de modo a justificar a identificação, transformação e extração desses dados para análise e suporte do processo de tomada de decisão (Liu et al., 2013). De acordo com Chandarana e Vijayalakshmi (2014), integrar diferentes tipos de dados, com vista a extrair informação para o negócio e provocar vantagens competitivas, representa o valor do *big data*. Além disso, um quinto "V" é utilizado para caracterizar *big data*: a veracidade, a qual tem relação com autenticidade, reputação de origem e confiabilidade dos dados, ou seja, esse critério classifica se a origem dos dados coletados é comprovada, se eles são confiáveis, se estão dentro da validade (ou vigência) e, quando os dados estão desatualizados, se são identificados ou tratados (Machado, 2017). A Figura 2 sintetiza os cinco atributos principais nos quais *big data* se baseia, representando, assim, o modelo dos 5 "Vs".

Figura 2: Os 5 "Vs" do *Big Data*.



Fonte: Adaptado de Chandarana e Vijayalakshmi, 2014.

Big Data tem potencial para desempenhar um papel de extrema relevância em várias áreas de aplicação, porém existem diversos desafios a ele associados, dentre os quais são citados por Chandarana e Vijayalakshmi (2014):

- Privacidade, segurança e confiabilidade;
- Gestão e compartilhamento de dados;
- Habilidades tecnológicas e analíticas;
- Armazenamento e processamento de dados;
- Propriedade dos dados, como crescimento, velocidade, escalabilidade e formatos.

Embora em constante evolução, os recursos computacionais convencionais são insuficientes para acompanhar a crescente complexidade do *big data*. As próximas seções visam explanar alguns padrões e termos associados ao *big data*, uma vez que alteram a maneira como se lida com o armazenamento e o processamento de grandes conjuntos de dados.

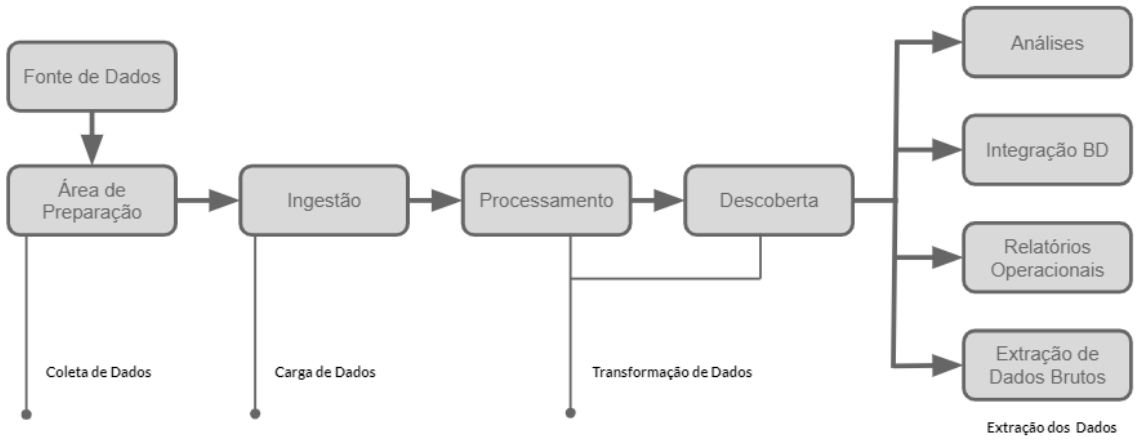
2.2.1. Processamento de *Big Data*

De acordo com Krishnan (2013), o processamento tradicional de dados pode ser definido como a coleta, o processamento e o gerenciamento de dados, resultando na geração de informações para os consumidores finais. Os dados transacionais são processados seguindo esse ciclo de vida, já que são estruturados por natureza e discretos em volume. Quando se trata de *big data*, o qual não é estruturado nem possui um volume finito, as complexidades de processamento aumentam. As arquiteturas de processamento tradicionais, como as plataformas *Symmetric Multi Processing* (SMP) ou *Massive Parallel Processing* (MPP), que são mais propensas a transações e orientadas a disco, não podem fornecer a escalabilidade, flexibilidade e *throughput* necessários ao *big data*. Segundo Krishnan (2013), em um ambiente tradicional, primeiro é feita a análise dos dados e a criação de um conjunto de requisitos, o que leva à descoberta de dados e à criação de modelos de dados e, em seguida, uma estrutura de banco de dados é criada para processar os dados. Já em um ambiente de *big data*, os dados são coletados e carregados numa determinada plataforma, depois os metadados são aplicados e, em seguida, cria-se a estrutura. Assim que a estrutura é aplicada, os dados são transformados e analisados. Portanto, para processar *big data*, uma arquitetura orientada a banco de dados seria inadequada. Para atender ao volume e a complexidade inerentes do *big data*,

uma arquitetura orientada a arquivos com uma *interface* de linguagem de programação se mostra mais adequada.

O fluxo de processamento de *big data* é descrito por Krishnan (2013) conforme observa-se na Figura 3, sendo possível identificar 4 fases:

Figura 3: Fluxo de Processamento de *Big Data.*



Fonte: Adaptado de Krishnan, 2013.

1) Coleta de Dados: Os dados são recebidos de diferentes fontes e carregados em um sistema de arquivos denominado área de preparação.
2) Carga de Dados: Os dados são carregados, aplicados metadados sobre eles e preparados para a transformação. Esse processo divide a entrada em pequenos pedaços de arquivos que podem ser particionados de maneira horizontal, ou seja, por conteúdo, ou vertical, por estrutura.
3) Transformação de Dados: Os dados são transformados, aplicando-se regras de negócios e processando o seu conteúdo. Esse estágio tem várias etapas a serem executadas e pode se tornar rapidamente complexo de gerenciar. As etapas de processamento em cada estágio produzem resultados intermediários, os quais geralmente são compostos por chaves de metadados e métricas associadas (um par de chave-valor).
4) Extração dos Dados: O conjunto de dados resultante pode ser extraído para processamento adicional, incluindo análises, relatórios operacionais, integração com *Data Warehouse* e propósitos de visualização.

White (2015) afirma que, embora as capacidades de armazenamento dos discos rígidos tenham aumentado enormemente ao longo dos anos, as velocidades de acesso não

se mantiveram. A maneira de se reduzir esse tempo é ler vários discos ao mesmo tempo. Porém, segundo White (2015), existem alguns problemas associados ao paralelismo que devem ser tratados. O primeiro é a falha de *hardware*, pois no momento em que vários componentes de *hardware* devem ser utilizados, aumenta a chance de um deles falhar. Uma maneira comum de evitar a perda é através de replicação. Outro problema é que a maioria das atividades de análise deve ser capaz de combinar os dados que estão distribuídos entre os vários discos e conseguir realizar isso é notoriamente desafiador.

Em um cenário em que a escalabilidade de aplicações e a necessidade de armazenar e processar grandes volumes de dados, muitas vezes não estruturados, acarretou desafios para as tecnologias tradicionais, emergem soluções que visam resolver os desafios impostos por um ambiente de *big data*, tais como sistemas de arquivos distribuídos, modelo de programação *MapReduce* e bancos de dados não relacionais, os quais serão brevemente abordados nos tópicos subsequentes.

2.2.2. Sistema de Arquivos Distribuído

A partir do momento em que um conjunto de dados excede a capacidade de armazenamento de uma única máquina física, torna-se necessário dividir esse conjunto de dados em máquinas separadas. Os sistemas de arquivos que gerenciam o armazenamento em uma rede de máquinas são chamados de sistemas de arquivos distribuídos (White, 2015). Coulouris et al. (2011) definem sistema de arquivos distribuídos como sendo aquele que permite aos programas armazenarem e acessarem arquivos remotos exatamente como se fossem locais, possibilitando que os usuários acessem arquivos a partir de qualquer computador em uma rede, mantendo o desempenho e a segurança comparáveis ao acesso de arquivos armazenados em discos locais.

Um sistema de arquivos distribuído possui as mesmas características de um sistema de arquivos convencional, porém ele deve permitir o armazenamento e o compartilhamento desses arquivos de forma distribuída em diferentes computadores, interconectados por meio de um *cluster*. Deve oferecer um desempenho similar ao de um sistema de arquivos tradicional e ainda prover escalabilidade. Além disso, existem algumas estruturas de controle exclusivas ou mais complexas que devem ser implementadas em um sistema de arquivos distribuído, dentre as quais citamos: Transparência, Tolerância a falhas, Integridade, Segurança, Desempenho e Consistência. Constituem exemplos de sistemas de arquivos distribuídos, dentre outros, o NFS

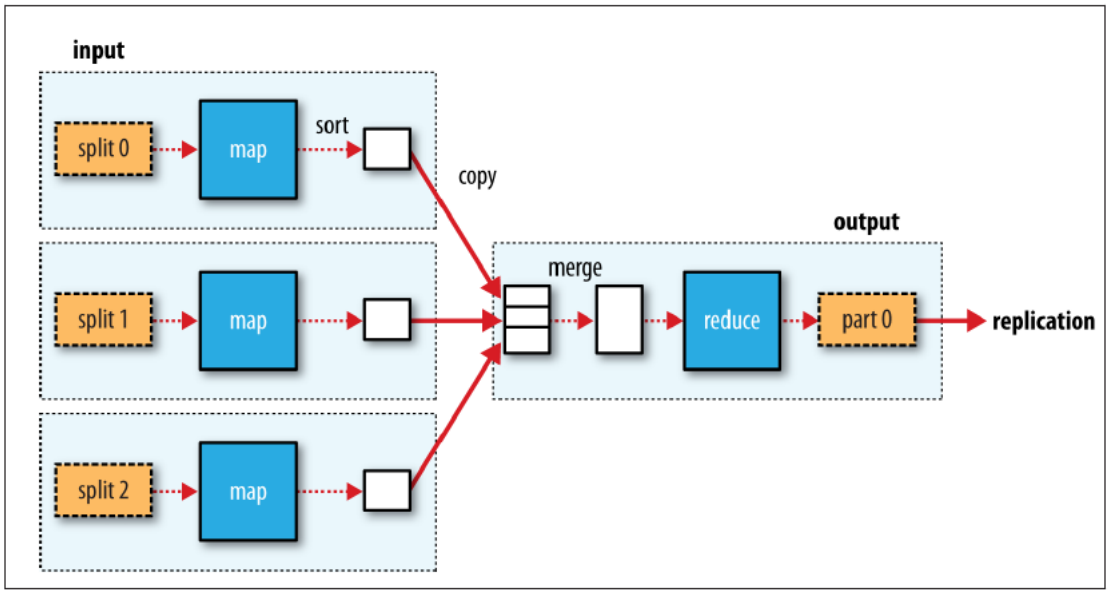
(*Network File System*), o GFS (*Google File System*) e o HDFS (*Hadoop Distributed File System*).

2.2.3. Modelo de Programação *MapReduce*

Diante de seu próprio conjunto de desafios únicos, em 2004 a *Google* decidiu trazer o poder da computação distribuída paralela para ajudar a digerir a enorme quantidade de dados produzidos durante as operações diárias. O resultado foi um grupo de tecnologias e filosofias de *design* arquitetônico que passaram a ser conhecidas como *MapReduce* (Schneider, 2012).

O *MapReduce* consiste em um modelo de programação para processamento de dados de forma paralela, aplicado especialmente para grandes conjuntos de dados (White, 2015). Funciona através de duas fases: a fase do mapa (*map*) e a fase de redução (*reduce*). Uma tarefa *MapReduce* divide o conjunto de dados de entrada em blocos independentes que são processados pelas tarefas do *map* de maneira completamente paralela. A estrutura classifica as saídas dos *maps*, as quais são inseridas nas tarefas de *reduce*. Normalmente, tanto a entrada quanto a saída da tarefa são armazenadas em um sistema de arquivos. Cada fase possui pares de chave-valor como entrada e saída, cujos tipos podem ser escolhidos pelo programador. O programador também especifica as funções *map* e *reduce*. A Figura 4 ilustra o fluxo de dados do *MapReduce* com uma única tarefa de redução.

Figura 4: Fluxo de Dados *MapReduce*.



Fonte: Adaptado de White, 2015.

De acordo com Schneider (2012), o *MapReduce* serviu de base para tecnologias subsequentes, como o *Hadoop*, enquanto o *Google File System* foi a base do *Hadoop Distributed File System* (HDFS), ambos abordados mais adiante neste capítulo.

2.3. Algoritmos de Aprendizado de Máquina Aplicados ao *Big Data*

O aprendizado de máquina (*Machine Learning*) é um segmento da Inteligência Artificial que está relacionado à questão de como construir programas de computador que melhoram automaticamente com a experiência (Mitchell, 1997). De acordo com Oladipupo (2010), tradicionalmente, é possível classificar os algoritmos de aprendizado de máquina, de acordo com o resultado desejado, nas categorias abaixo:

- Supervisionado: Os algoritmos usam um conjunto de dados rotulado por seres humanos para prever o resultado desejado e conhecido. Normalmente utilizado para problemas de classificação e regressão.
- Não Supervisionado: Algoritmos que trabalham com dados não rotulados, buscando encontrar parâmetros comuns e agrupá-los de modo a encontrar estruturas que não eram conhecidas.
- Por Reforço: Algoritmos que buscam desenvolver uma política de ações com maior ganho em um dado ambiente, utilizando, para isso, as entradas fornecidas para explorar o ambiente e analisam os impactos nos possíveis caminhos para se orientar.

Existe um vasto conjunto de algoritmos de aprendizado de máquina disponíveis para o desenvolvimento de sistemas preditivos inteligentes aplicados a diversas áreas, pois essas técnicas fornecem soluções possíveis para extrair as informações ocultas nos dados. Entretanto, com o advento do *big data*, os conjuntos de dados se tornam tão grandes e complexos que é difícil utilizar os métodos tradicionais de aprendizado, já que tais métodos não foram projetados para trabalhar com altos volumes de dados (Qiu, J. et al., 2016).

Em Qiu, J. et al. (2016) os principais problemas de aprendizado de máquina para *big data* e as possíveis soluções são sintetizados na Tabela 1:

Tabela 1: Principais Problemas do Aprendizado de Máquina para *Big Data* e Possíveis Soluções.

Problemas	**Possíveis Soluções**
Grande escala de dados.	Estruturas distribuídas com computação paralela, como ADMM (*Alternating direction method of multipliers*), modelo de programação *MapReduce* e Computação em Nuvem.
Diferentes tipos de dados.	Aprendizado por representação; *Deep Learning*; Redução de Dimensionalidade; Aprendizado baseado em *kernel*.
Alta velocidade de dados.	Aprendizado *online*; Processamento *Streaming*, como Borealis, S4 e Kafka.
Dados incertos e incompletos.	Aplicar estatísticas resumidas, como médias e variâncias para abstrair as distribuições de amostras; Utilizar informações completas transportadas pelas distribuições de probabilidade para construir uma árvore de decisão.
Dados de baixa densidade de valor e diversidade de significado.	Tecnologias de mineração de dados e KDD (*Knowledge Discovery in Databases*); Tecnologias de Aprendizagem Cognitiva.

Fonte: Adaptado de Qiu, J. et al., 2016.

Em resumo, os aspectos tratados na tabela acima refletem as principais características de *big data*, ou seja, volume, variedade, velocidade, veracidade e valor, sobre as quais conferem diferenças significativas nos métodos de aprendizado de máquina, sendo necessários métodos escalonáveis, com múltiplos domínios, paralelos, flexíveis e inteligentes, sendo ainda necessárias várias tecnologias facilitadoras para integrar o processo de aprendizagem e torná-lo eficaz (Qiu, J. et al., 2016).

Nas seções 2.3.1 e 2.3.2 serão apresentados dois exemplos de algoritmos de aprendizado de máquina que podem ser aplicados no contexto desta dissertação, considerando que o trabalho visa a classificação de notícias, tendo como categoria mais adequada a de algoritmos supervisionados, e considerando ainda as limitações existentes no pacote utilizado no desenvolvimento deste trabalho, limitações estas que serão detalhadas no Capítulo 6.

2.3.1. Naive Bayes

De acordo com Han, Kamber e Pei (2011), *Naive Bayes* é um classificador probabilístico baseado no Teorema de *Bayes*, formulado matematicamente por Thomas Bayes em 1763, para determinar a classe de maior probabilidade para cada nova instância a ser classificada. Esse classificador pode prever probabilidades de associação à classe, como a probabilidade de que uma determinada tupla pertença a uma classe específica. Classificadores *Naive Bayes* supõem que o efeito de um valor de atributo em uma determinada classe é independente dos valores dos outros atributos, suposição esta denominada independência condicional.

No Teorema de *Bayes*, seja B uma tupla de dados, considerada "evidência", e A uma hipótese como a de que a tupla B pertença a uma classe específica C, queremos determinar P(A|B), ou seja, determinar a probabilidade de que a tupla B pertença a classe C, pois sabemos a descrição do atributo B. P(A|B) é a probabilidade posterior, ou *a posteriori*, de A condicionada em B e é representada pela equação 2.1. P(A) é a probabilidade anterior, ou *a priori*, de A.

$$\boldsymbol{P(A|B) = \frac{P(B|A)P(A)}{P(B)}} \tag{2.1}$$

O Teorema de *Bayes* representa uma combinação dos teoremas da probabilidade condicional e da probabilidade total, conforme mostrado na equação 2.2.

$$\boldsymbol{P(B_i|A) = \frac{P(A|B_i)P(B_i)}{\sum_{j=1}^{n}[P(A|B_j)P(B_j)]}} \tag{2.2}$$

Para classificar uma nova instância, o algoritmo determina a classe mais provável, dados os atributos (a_1, a_2, ..., a_n) que descrevem a instância (Mitchell, 1997). A equação 2.3 mostra o cálculo da classe de maior probabilidade para o classificador *Naive Bayes*, onde V_{NB} representa a resposta do classificador, $P(v_j)$ diz respeito a frequência estimada de instâncias de treinamento que pertencem a cada classe v_j, e $P(a_i|v_j)$ é a frequência estimada dos valores do atributo a_i restrito aos exemplos de treinamento das classes v_j.

$$\boldsymbol{V_{NB} = \underset{v_j \in V}{\arg\max} P(v_j) \prod_i P(a_i|v_j)} \tag{2.3}$$

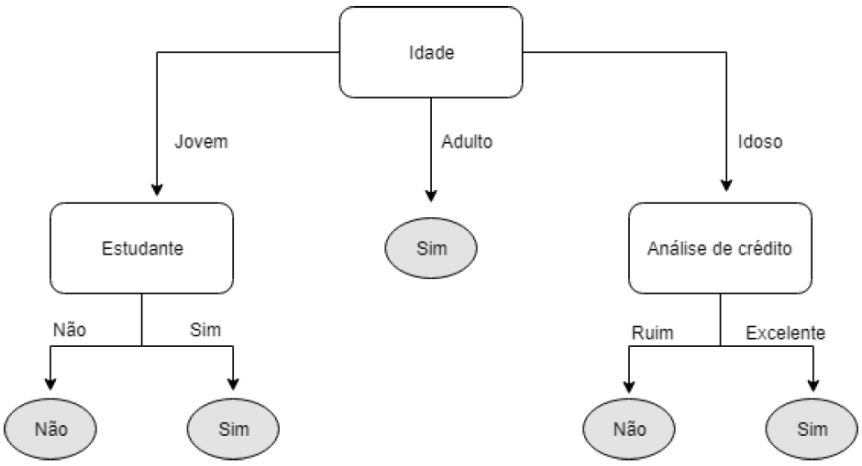
Segundo Haindl et al. (2006), apesar de toda a matemática envolvida, implementar um classificador *Bayes* é como contar o número de palavras, documentos e categorias, ou seja, após avaliar o número de palavras positivas e negativas em uma frase, ele pode ser combinado para calcular a probabilidade de cada uma das classes possíveis. O documento é então classificado de acordo com a maior probabilidade calculada.

Os classificadores *Bayesianos* têm sido aplicados para resolver problemas em diversas áreas de pesquisa, tais como processamento de linguagem natural, diagnósticos médicos, busca heurística, questões ambientais, gestão de bacias hidrográficas, entre outras (Borunda et al., 2016). Essa gama de possibilidades de aplicação torna evidente a eficiência e versatilidade desse classificador em diversos cenários.

2.3.2. Árvore de Decisão e *Random Forest*

Uma árvore de decisão é uma estrutura de árvore do tipo fluxograma, em que cada nó interno (nó não-folha) denota um teste em um atributo, cada ramificação representa um resultado do teste e cada nó folha (ou nó terminal) possui um rótulo de classe. O nó mais alto de uma árvore é o nó raiz (Han; Kamber; Pei, 2011). A Figura 5 apresenta um exemplo de árvore de decisão para prever se um cliente irá comprar um computador. Cada nó interno representa um teste em um atributo e cada nó folha representa uma classe. Existem três atributos que influenciam na classificação deste exemplo: Idade, Estudante e Análise de Crédito, e duas classes: Sim ou Não.

Figura 5: Exemplo de Árvore de Decisão.



Fonte: Adaptado de Hans, Kamber e Pei, 2011.

O conjunto de dados disponível é dividido em conjunto de dados de treinamento e conjunto de dados de validação. O conjunto de dados de treinamento é usado para construir um modelo de árvore de decisão e o conjunto de dados de validação é usado para decidir sobre o tamanho apropriado da árvore necessário para alcançar o modelo final ideal. Ao construir uma árvore de decisão, um dos atributos é selecionado para atuar como um nó raiz. Essa escolha ocorre através de medidas de seleção.

Existem diversos algoritmos para implementação das árvores de decisão. Os algoritmos mais utilizados são ID3, C4.5 e CART. De acordo com Han, Kamber e Pei (2011), o algoritmo ID3 utiliza o ganho de informação para selecionar a melhor divisão. É possível construir uma árvore de decisão através de um conjunto de dados categóricos somente, sendo essa uma das suas principais limitações, além de não apresentar uma forma para tratar valores desconhecidos. O algoritmo C4.5 foi um sucessor do ID3, possuindo como principais diferenças a capacidade de lidar com dados tanto categóricos quanto contínuos, de tratar valores desconhecidos (instâncias sem rótulo), de utilizar a medida de razão de ganho para selecionar o atributo que melhor divide os exemplos (medida essa que se mostrou superior ao ganho de informação) e de apresentar um método de pós-poda moldando a estrutura da árvore de baixo para cima, melhorando o seu desempenho. Por fim, o algoritmo CART (*Classification and Regression Trees*) consiste em uma técnica que produz árvores binárias que permitem manipular dados categóricos e contínuos e possui uma função pós-poda por meio da redução do fator custo-complexidade.

O algoritmo *Random Forest* é baseado no conceito de árvore de decisão, porém, em contrapartida a geração de uma única árvore, tal método produz múltiplas árvores de decisão, sendo, portanto, denominadas de Floresta (*Forest*). O resultado das árvores é combinado, a partir de votações e é escolhido o modelo com maior quantidade de votos (Alpaydin, 2016).

Conforme detalhado em Han, Kamber e Pei (2011), a floresta normalmente é criada pelo método *bagging*, cuja ideia principal é agregar vários preditores, produzindo um preditor mais estável. O algoritmo adiciona aleatoriedade ao modelo, pois ao invés de buscar pelo melhor atributo ao fazer a divisão de nós, ele busca o melhor atributo em um subconjunto aleatório de atributos. Este processo cria uma grande diversidade, o que geralmente leva a geração de modelos melhores.

Dentre as vantagens da utilização de *Random Forest*, destacamos a capacidade de lidar com dados em grandes volumes e com muitas dimensões, sendo possível tratar milhares de variáveis de entrada e identificar as mais significativas, além de ser eficaz em estimar dados faltantes e manter a precisão mesmo com muitos dados desconhecidos.

2.4. Tecnologias de Big Data

Conforme discutido anteriormente, o modelo de computação paralela e distribuída possui atualmente papel fundamental no processamento e na extração de informação relevante das aplicações de *big data*. Porém, devido a complexidade inerente a esse modelo, algumas das suas características inibem sua utilização por novos usuários. Dividir uma tarefa em subtarefas e então executá-las paralelamente em diversas unidades de processamento não é algo trivial. Se o tamanho e a divisão das tarefas não forem bem dimensionados, isso pode comprometer totalmente o desempenho da aplicação. Além disso, é necessário extrair a dependência entre os dados da aplicação, determinar um algoritmo de balanceamento de carga e de escalonamento para as tarefas, garantir a eficiência no uso dos recursos computacionais e a recuperação (ou a não interrupção da execução) da aplicação caso uma máquina falhe.

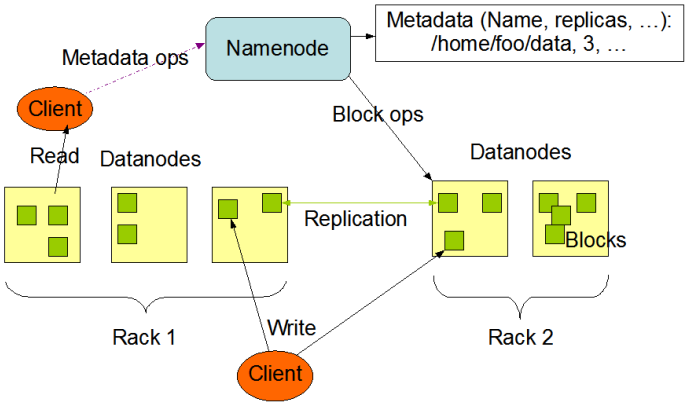
Nesse contexto foi desenvolvido o *Hadoop*, um *framework* que permite processamento de grandes conjuntos de dados em *clusters* de computadores (Hadoop, 2019). O *Hadoop* é mantido pela *Apache Foudation*, mas foi fruto da colaboração de algumas das maiores empresas do mundo, como *IBM*, *Microsoft*, *Amazon* e *Oracle* (White, 2015). O projeto é implementado em *Java* e inspirado no trabalho desenvolvido pela *Google*, no *Google File System* e no paradigma de programação *MapReduce*. Apesar de possuir como elementos-chave o sistema de arquivos distribuído, denominado *HDFS*, e o modelo de programação *MapReduce*, em meio a grande evolução que esse *framework* teve, novos subprojetos foram incorporados ao que se chamou de Ecossistema *Hadoop*, cada um com uma proposta específica de aplicação, tornando a infraestrutura do *framework* cada vez mais completa.

Abaixo discutimos algumas das ferramentas que compõem o ecossistema *Hadoop* e que foram utilizadas na arquitetura proposta por esta dissertação.

HDFS: O *Hadoop Distributed File System* corresponde ao componente principal do ecossistema *Hadoop* (Sinha, 2019). A arquitetura do *HDFS* é do tipo mestre-escravo,

onde do lado mestre existe um nó central, denominado *NameNode*, o qual tem a função de armazenar os metadados dos arquivos. E do lado escravo, existem diversas máquinas denominadas *DataNodes*, as quais armazenam os dados propriamente ditos e realizam o processamento dos dados. O *HDFS* replica os blocos para aumentar a segurança. Antes de enviar os blocos aos nós escravos, cada um dos arquivos é subdividido em n blocos de tamanho fixo de 64MB (Apache, 2019b). A Figura 6 mostra um modelo de arquitetura do *HDFS*, onde existem cinco *DataNodes* divididos em dois *Racks* e um *NameNode*. O cliente acessa o *NameNode* para consultar onde os blocos de um determinado arquivo estão armazenados ou onde armazenar novos blocos e, após informação da localização, o cliente acessa diretamente o *DataNode* para obter ou escrever os blocos.

Figura 6: Arquitetura HDFS.



Fonte: Apache, 2019b.

MapReduce: Consiste em um *framework* para escrever aplicativos que processam grandes quantidades de dados (conjuntos de dados de vários *terabytes*) em paralelo em grandes clusters de *hardware* de *commodity* de maneira confiável e tolerante a falhas (Apache, 2019c). É composto por um *JobTracker* no nó mestre, responsável por agendar as tarefas dos escravos, monitorando-as e reexecutando aquelas com falha, e vários *Tasktracker* nos nós escravos, os quais são responsáveis por executar as tarefas propriamente ditas, conforme instrução do mestre (Apache, 2019c).

HBase: É um banco de dados não relacional de código aberto e distribuído (Apache, 2019e). Fornece armazenamento de dados paralelo por meio dos sistemas de arquivos distribuídos subjacentes entre os servidores comuns. O sistema de arquivos de escolha é

tipicamente o *HDFS*, devido forte integração do *HBase* e do *HDFS* (Lee et al., 2013). Faz parte da camada de armazenamento da pilha de projetos *Hadoop*.

Hive: É um *software* de *data warehouse* que facilita a leitura, gravação e gerenciamento de grandes conjuntos de dados que residem no armazenamento distribuído, usando SQL (White, 2015). É considerado um padrão para consultas baseadas em *SQL* sobre grandes conjuntos de dados usando o *Hadoop* e oferece fácil extração de dados, transformação e acesso ao *HDFS*, incluindo arquivos de dados ou outro sistema de armazenamento *HBase* (Lee, 2013).

Mahout: É um *framework* de álgebra linear distribuída projetado para permitir que matemáticos, estatísticos e cientistas de dados implementem rapidamente seus próprios algoritmos (Apache, 2019d). Utiliza o *MapReduce* para construir algoritmos complexos de aprendizado de máquina aplicados ao campo da análise de dados em larga escala. As bibliotecas existentes no *Mahout* dividem-se na implementação de soluções para três principais assuntos do aprendizado de máquina: recomendação, classificação e *clustering*. O *Mahout* foi desenvolvido visando a escalabilidade e eficiência, convertendo seus algoritmos para trabalhar e um cluster *Hadoop*. É um projeto em desenvolvimento que, embora já possua muitas técnicas e algoritmos implementados, tem componentes em desenvolvimento ou em fase de testes.

2.5. Considerações Finais

Este capítulo apresentou o levantamento do estado da arte das técnicas e tecnologias envolvidas neste trabalho e que embasaram o desenvolvimento desta pesquisa. No próximo capítulo serão apresentados os trabalhos relacionados ao estudo realizado.

3. TRABALHOS CORRELATOS

3.1. Considerações Iniciais

Nesta seção será apresentado um levantamento dos trabalhos correlatos pesquisados e selecionados que fazem uso de tecnologias de *big data* em diferentes áreas, mostrando as ferramentas e técnicas aplicadas, e traz algumas abordagens que se relacionam com o objetivo desta proposta de trabalho.

3.2. Correlatos

Desde o advento do conceito de *big data*, algumas arquiteturas para gerenciar e analisar estes dados em diferentes campos foram propostas, tendo suas bases técnicas em paradigmas de computação distribuída, como computação em grade (Berman et al., 2003). Contudo, a atual "explosão" de dados, na qual há uma geração diária de grandes quantidades de dados de vários formatos e fontes, está revelando o significado mais completo de *big data*. Portanto, presenciamos o surgimento constante de novas tecnologias construídas para serem aplicadas neste contexto.

No trabalho de Wu, L.; Feng, X.; Shen, Z. (2016), é introduzida uma nova arquitetura, baseada no ecossistema *Hadoop*, para sistemas de gerenciamento de grupos de empresas elétricas, a fim de romper os gargalos de desempenho do sistema tradicional. A arquitetura nova foi composta pelos módulos já existentes na arquitetura tradicional, adicionando-se um módulo de processamento distribuído baseado no *Hadoop*. Esta arquitetura foi validada, através de sua implementação e experimentação, garantindo a melhora efetiva no desempenho do processamento de dados corporativos e a eficiência do sistema de gerenciamento do grupo de energia elétrica.

No domínio da saúde, uma arquitetura de *big data* completa e específica foi desenvolvida por Wang et al. (2016). Essa arquitetura foi baseada nas experiências sobre as melhores práticas na implementação de sistemas de *big data* na indústria e foi composta por cinco camadas principais: a camada de dados, que inclui as fontes de dados a serem usadas no suporte às operações e resolução de problemas; a camada de agregação de dados, encarregada de adquirir, transformar e armazenar dados; a camada de análise, responsável pelo processamento e análise dos dados; a camada de exploração da informação, que funciona gerando saídas para o apoio à decisão clínica, como o

monitoramento em tempo real de possíveis riscos médicos; por último, a camada de governança de dados, encarregada de gerenciar os dados comerciais durante todo o ciclo de vida, aplicando os padrões e políticas adequados de segurança e privacidade. Esta camada é particularmente necessária neste caso, dada a sensibilidade dos dados clínicos.

Yuvaraj, N.; Sripreethaa K.R. (2017) apresentaram uma aplicação para previsão de *diabetes* usando três algoritmos de *Machine Learning* baseados em um *cluster Hadoop*: *Random Forest*, Árvore de Decisão e *Naive Bayes*. O conjunto de dados *Pima Indian Diabetes* (PID) foi utilizado após o pré-processamento. Os autores não mencionaram como os dados foram pré-processados, mas discutiram o método de ganho de informações usado para a seleção de recursos para extrair os mais relevantes. Eles usaram oito atributos principais e dividiram o conjunto de dados em 70% para treinamento e 30% para teste. Os resultados mostraram que o algoritmo *Random Forest* teve a maior taxa de precisão de 94%.

Ainda no contexto da saúde, em Kumar, S.; Singh, M. (2019) é realizada uma pesquisa sobre o impacto do *big data* nos serviços de saúde e as várias ferramentas disponíveis no ecossistema *Hadoop* para tratá-lo. Também explora-se uma arquitetura conceitual da análise de *big data* para saúde, que envolve o histórico de coleta de dados de diferentes fontes, o banco de dados do genoma, registros eletrônicos de saúde, imagem/texto e sistemas de apoio à decisão clínica.

A pesquisa de Revathy, R. et al. (2019) utiliza o *Hadoop* para classificar as pragas das culturas agrícolas. A base de dados é primeiramente pré-processada e então é utilizado o algoritmo de árvore de decisão C5.0 baseado no *Hadoop MapReduce* para a classificação das pragas. Como o tamanho do conjunto de dados de pragas atingiu a faixa de *terabytes*, as técnicas típicas de mineração de dados não podem processar o *big data* em tempo adequado. O sistema de arquivos HDFS e o modelo de programação *MapReduce* foram colocados em prática para realizar o armazenamento e processamento distribuídos dessa grande quantidade de dados.

No âmbito das *fake news*, Pérez-Rosas, V. et al (2018) criou uma base de dados de notícias verdadeiras e falsas e, em seguida, utilizou um classificador SVM linear para distinguir esses pares de histórias entre notícias reais e falsas. Esse algoritmo realizou então uma análise de notícias retiradas da *web*, obtendo uma taxa de sucesso de 76% no

melhor caso. Já na pesquisa de Monteiro, R. A. et al (2018) investiga-se a detecção de notícias falsas para a língua portuguesa, introduzindo o primeiro *corpus* de referência nesta área para o português, contendo notícias verdadeiras e falsas e aplicando-se técnicas tradicionais de aprendizado de máquina, com obtenção de bons resultados. Em ambas as pesquisas não são utilizadas ferramentas de *big data*, sendo trabalhados, portanto, somente com dados de um único formato.

Todos os trabalhos apresentados possuem contribuições na área desta dissertação e, em alguma medida, colaboraram para a construção desta pesquisa. A Tabela 3 abaixo descreve as lacunas dos principais trabalhos abordados neste capítulo.

Tabela 2: Síntese dos Trabalhos Pesquisados.

Autores	**Principais Lacunas Encontradas**
Wu, L.; Feng, X.; Shen, Z. (2016)	O trabalho utiliza SGBD proprietário durante os experimentos, o que dificulta a reprodução da solução. Ademais, não foram aplicadas otimizações ao modelo experimental, como a utilização de processamento *in-memory*, para fins de comparação com os experimentos realizados no trabalho.
Wang et al. (2016)	Apesar de propor uma arquitetura de *big data* completa, não foram realizados experimentos para validação da solução proposta.
Yuvaraj, N.; Sripreethaa K.R. (2017)	A base de dados utilizada no trabalho é composta apenas por dados estruturados que, apesar de volumosos, não contempla totalmente a definição de *Big Data*.
Pérez-Rosas, V. et al (2018)	O *corpus* utilizado contém notícias exclusivamente na língua inglesa, adequado

	obviamente ao contexto em que o trabalho foi aplicado, não sendo apropriado para o contexto de notícias na língua portuguesa. Além disso, utiliza dados em apenas um único formato, não sendo aplicado para os propósitos de *big data*.
Monteiro, R. A. et al (2018)	Utilizam apenas o formato textual para análise de notícias, não sendo considerados tipos distintos de dados, ou seja, não sendo aplicável em um contexto de *big data*.
Kumar, S.; Singh, M. (2019)	Não há implementação da arquitetura proposta para validação da solução.
Revathy, R. et al. (2019)	O trabalho não realiza experimentos com algoritmos de aprendizado de máquina disponíveis no pacote *Mahout*, desenvolvido para adequada utilização dentro de um *cluster Hadoop*. Também não são utilizadas outras tecnologias que podem otimizar o resultado da proposta em questão, como o processamento *in-memory*.

Fonte: Autor, 2019.

3.3. Considerações Finais

Pelo estudo dos trabalhos expostos neste capítulo, observa-se que há um conjunto de pesquisas sendo realizadas no sentido de utilizar ferramentas e técnicas de *big data* em vários contextos. Não foram encontrados trabalhos publicados relacionados com a utilização de uma arquitetura de *big data* no contexto de *fake news*.

De modo geral, os trabalhos mencionados identificaram os distintos e relevantes componentes de uma arquitetura de *big data*. A arquitetura proposta nesta dissertação considera tais componentes consolidados na literatura. Desta forma a pesquisa contribui para o armazenamento, processamento e classificação de notícias, com o propósito de

auxiliar na verificação de sua veracidade. A consolidação da proposta deste trabalho baseia-se em validar a utilização de tecnologias de *big data* consistentes com a literatura, aplicadas ao propósito da problemática em questão, no contexto nacional, realizando experimentos para tal validação. Nesta pesquisa, explora-se o potencial para o desenvolvimento de uma arquitetura de *big data* voltada para a classificação de notícias falsas com o propósito de expandir as soluções atualmente utilizadas na detecção de *fake news*, possibilitando a avaliação de um volume expressivo de notícias em formatos heterogêneos.

4. ARQUITETURA DE *BIG DATA* PROPOSTA

4.1. Considerações Iniciais

Neste capítulo é apresentada a proposta de arquitetura de *Big Data* para auxiliar na detecção de *Fake News*, a qual foi dividida em camadas, incluindo as tecnologias e ferramentas utilizadas em cada camada aplicadas ao presente trabalho.

4.2. Arquitetura Proposta

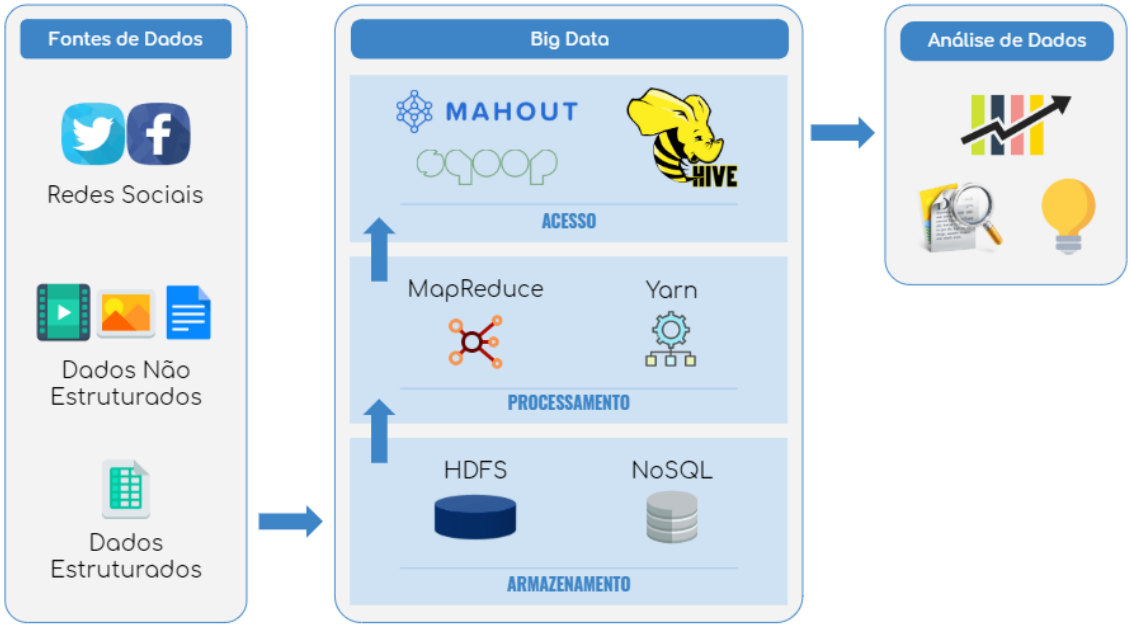
Uma arquitetura de *big data* descreve uma abordagem de dimensões para lidar com a viabilidade de uma solução de *big data* (Mysore; Khupat; Jain, 2014). É comumente projetada para lidar com a extração, processamento e análise de dados grandes e complexos demais para sistemas de armazenamento e processamento tradicionais. Normalmente é dividida em camadas e composta por um conjunto de componentes com propósitos específicos que são interligados entre si para produzir resultados satisfatórios no contexto de *big data*.

Com a grande quantidade de notícias sendo gerada diariamente por diversas fontes, produzindo grandes conjuntos de dados em alta velocidade, as tecnologias de bancos de dados relacionais tornam-se inadequadas para lidar com essa massiva quantidade de dados, devido a velocidade de processamento limitada e o custo significativo de expansão de armazenamento. Para resolver esse problema, as tecnologias de *big data* surgem como alternativa, pois são capazes de analisar grandes bases de dados, processar cálculos pesados, identificar comportamentos e disponibilizar serviços especializados. Para construir uma arquitetura capaz de alcançar os objetivos desta proposta de trabalho, alguns requisitos tornam-se necessários e foram considerados para a composição da arquitetura:

- Extrair dados de fontes diversas.
- Utilizar formatos distintos de dados (estruturados e não estruturados).
- Armazenar e processar os dados de maneira distribuída.
- Utilizar algoritmos de aprendizado de máquina para classificar notícias.
- Permitir o uso de ferramentas para análise dos resultados.

A estrutura de *big data* proposta para auxiliar na detecção de *fake news* é dividida em camadas, conforme ilustra a Figura 7. Os detalhes de cada camada são apresentados nas seções seguintes.

Figura 7: Arquitetura de *Big Data* para Detecção de *Fake News*.

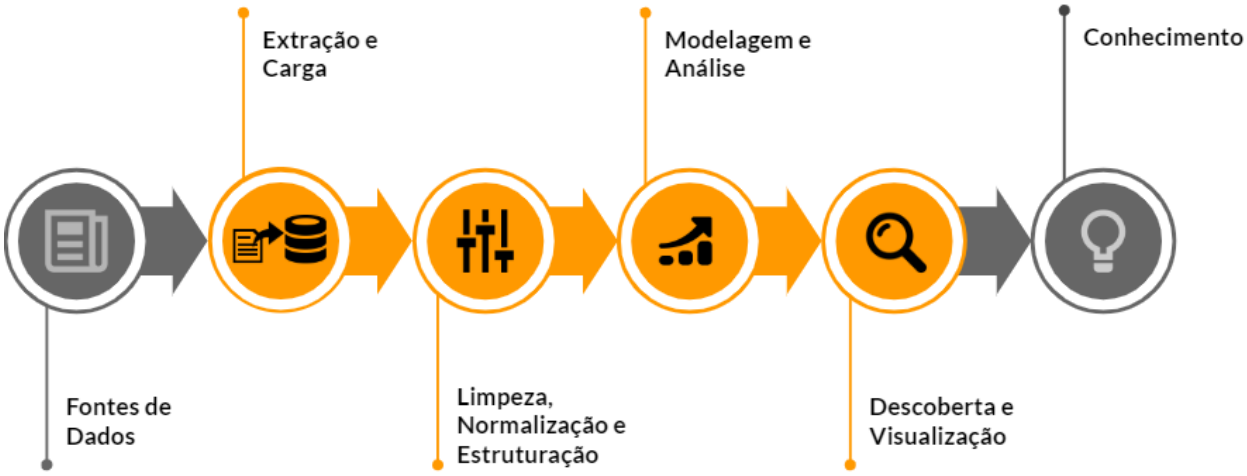


Fonte: Autor, 2019.

4.2.1. Camada de Fonte de Dados

A primeira camada da arquitetura, denominada camada de Fonte de Dados, visa aglomerar os diferentes tipos de dados a serem utilizados no projeto, bem como a origem destes dados. Nos cenários de *big data*, existem várias fontes de dados e, no que tange o trabalho em questão, podemos destacar duas principais, os *websites* de notícias e as redes sociais, de onde são extraídos formatos variados, ou seja, dados estruturados e não estruturados. Para este último tipo de dados destacamos os textos, que compõem a maior parte das notícias, as imagens e os vídeos. O fluxo dos dados é definido através das etapas ilustradas na Figura 8 e descritas em seguida.

Figura 8: Fluxo de Dados.



Fonte: Autor, 2019.

Para a extração e carga dos dados, podem ser determinadas duas formas, conforme mostra a Tabela 4.

Tabela 3: Formas de Extração e Carga de Dados.

Download de notícias.	Através de algoritmo extraem-se notícias atualizadas de *websites* pré-selecionados para compor o *corpus*, armazenado dentro da infraestrutura designada para o armazenamento dos dados, seja em um sistema de arquivos diretamente, seja através de um banco de dados relacional ou não relacional, dependendo do formato e volume dos dados.
Carga agendada.	Os dados são extraídos e carregados em um intervalo de tempo pré-definido para fins de atualização das bases de dados. Esse formato é aplicável especialmente para mídias sociais, cuja geração de dados ocorre em alta velocidade.

Fonte: Autor, 2019.

Após a carga de dados, são realizados alguns tratamentos para manter o alinhamento e padronização dos dados. Esses tratamentos devem levar em consideração os aspectos relacionados por Rubin, Chen e Conroy (2015): as notícias deve estar no formato texto sem formatação, pois geralmente é mais apropriado para a utilização do método de estruturação denominado NLP (*Natural Language Processing*); as notícias devem possuir tamanho semelhante em número de palavras para evitar distorções de aprendizado; um período de tempo deve ser especificado para a coleta de notícias, pois o

estilo de escrita pode mudar no decorrer do tempo; manutenção de fatores pragmáticos, como o *link* original das notícias, já que essas informações podem ser úteis no futuro para verificação de fatos. A NLP consiste em uma área de pesquisa focada na exploração de como os computadores podem ser usados para entender e modelar o texto em linguagem natural, para que possa ser útil para diferentes aplicações (Chowdhury, 2005). Compreende várias técnicas para estruturar textos de maneiras diferentes, para que as informações subjacentes possam ser extraídas com mais facilidade. Dentre essas técnicas estão: Análise de Sentimentos, Análise Semântica Latente (LSA) e TF-IDF (*Term Frequency – Inverse Document*).

Com os dados devidamente tratados, é o momento de realizar a modelagem e análise. Isso inclui a redução de dimensionalidade dos conjuntos de dados, aplicação de técnicas de modelagem aos dados e obtenção de resultados. A utilização de paradigmas de modelagem dos dados depende do tipo de dado e do objetivo da análise. Para esta pesquisa, como o objetivo é a classificação de notícias, foi utilizado o aprendizado Supervisionado. As demais etapas do fluxo de dados serão tratadas na camada de Análise de *Big Data*.

4.2.2. Camada de Armazenamento

Na camada de armazenamento são determinados os principais repositórios para o adequado funcionamento da solução. Nesta etapa, os dados podem ser armazenados em estruturas tecnológicas diferentes, dependendo do seu tipo e objetivos. Na presente pesquisa, optou-se pela utilização de uma estrutura de armazenamento distribuída, um banco de dados não relacional e um banco de dados relacional. Como estas tecnologias foram destinadas a propósitos distintos, na Figura 9 são apresentados os tipos de dados associados a cada tecnologia escolhida.

Figura 9: Tecnologias de Armazenamento.

Fonte: Autor, 2019.

O sistema de arquivos distribuído, *HDFS*, é projetado para armazenar arquivos grandes com padrões de acesso a dados de *streaming*, rodando em *hardware* comum (White, 2015). De maneira resumida, os aspectos considerados para a utilização do *HDFS* podem ser classificados conforme abaixo:

- Arquivos grandes: arquivos com *megabytes*, *gigabytes* ou *terabytes* de tamanho.
- *Streaming* de acesso a dados: utiliza o padrão de processamento de dados *write-once, read-many-times* (escreva uma vez, leia várias vezes), ou seja, um conjunto de dados é copiado de sua fonte de dados uma única vez e depois várias análises são aplicadas ao longo do tempo. Desta forma, o tempo para ler todo o conjunto de dados é mais importante do que a latência na leitura do primeiro registro.
- *Hardware* comum: utilização de *hardware* acessível, cuja chance de falha é alta quanto maior for o *cluster*, mas que mantém o trabalho de forma imperceptível ao usuário mediante tal falha.

É importante ressaltar que o *HDFS* não funciona bem para aplicativos que requerem acesso de baixa latência a dados, com arquivos pequenos e com múltiplos gravadores de dados. Para os casos em que é necessário acesso aleatório de leitura/escrita em grandes conjuntos de dados, a opção encontrada foi a utilização de um banco de dados distribuído orientado a coluna, o *HBase*, o qual é construído sobre o *HDFS*. Neste banco de dados não relacional podem ser armazenados dados históricos e os resultados das análises realizadas sobre os conjuntos de notícias.

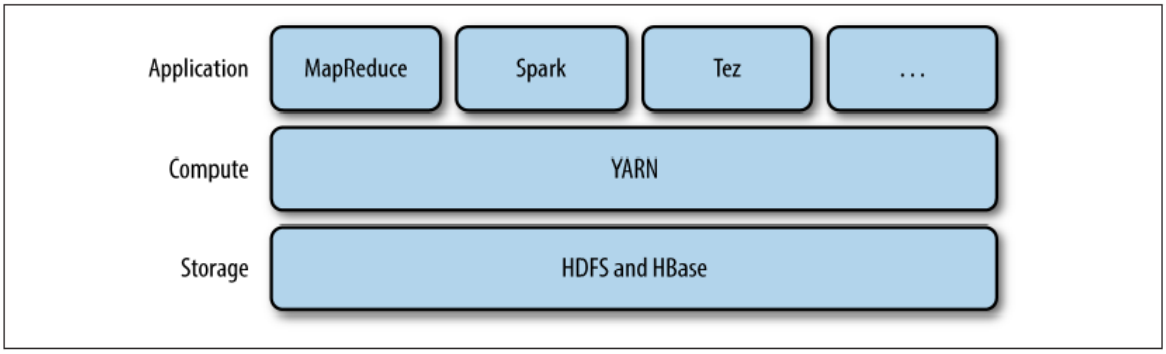
Por fim, para os arquivos texto pequenos (com menos de 1MB de tamanho) e sem qualquer tipo de tratamento, além de resultados de análises que são consultadas de maneira recorrente, é possível a utilização de um banco de dados relacional ou o *Hive*. Para este trabalho, após tratamento dos dados, optou-se por realizar o armazenamento de todo o conjunto de dados a ser utilizado na modelagem dentro de uma estrutura de diretórios diretamente no *HDFS*.

4.2.3. Camada de Processamento

Uma vez que os dados são armazenados nas diferentes tecnologias citadas no tópico 4.2.2, estes dados são processados, através da camada de processamento da arquitetura proposta, utilizando o modelo de programação *MapReduce*. Esse modelo de programação é inerentemente paralelo e permite executar programas que analisam grandes conjuntos de dados dentro de um *cluster Hadoop*, escritos em várias linguagens de programação diferentes, tais como *Java*, *Python* e *Ruby*. Para o projeto em questão, o *MapReduce* foi utilizado através das ferramentas contidas na camada de acesso, uma vez que tais ferramentas facilitam o acesso à análise e consulta de dados, por meio de códigos mais simples os quais são convertidos em *jobs MapReduce*.

Nesta mesma camada foi proposta a utilização do *YARN* (*Yet Another Resource Negotiator*) responsável por realizar o gerenciamento dos recursos do *cluster Hadoop*. Segundo White (2015), o *YARN* fornece *APIs* (*Application Programing Interfaces*) para requisitar e trabalhar com os recursos do *cluster*, mas sem utilização direta pelo código do usuário. Em vez disso, os usuários gravam em *APIs* de nível superior fornecidas por estruturas de computação distribuída, que são construídas no *YARN* e ocultam os detalhes do gerenciamento de recursos do usuário. Esta situação é ilustrada na Figura 10, onde é apresentada uma camada intermediária entre a camada de aplicação e de armazenamento, denominada camada de computação, demonstrando as principais aplicações suportadas pelo *Yarn*.

Figura 10: Aplicações YARN.



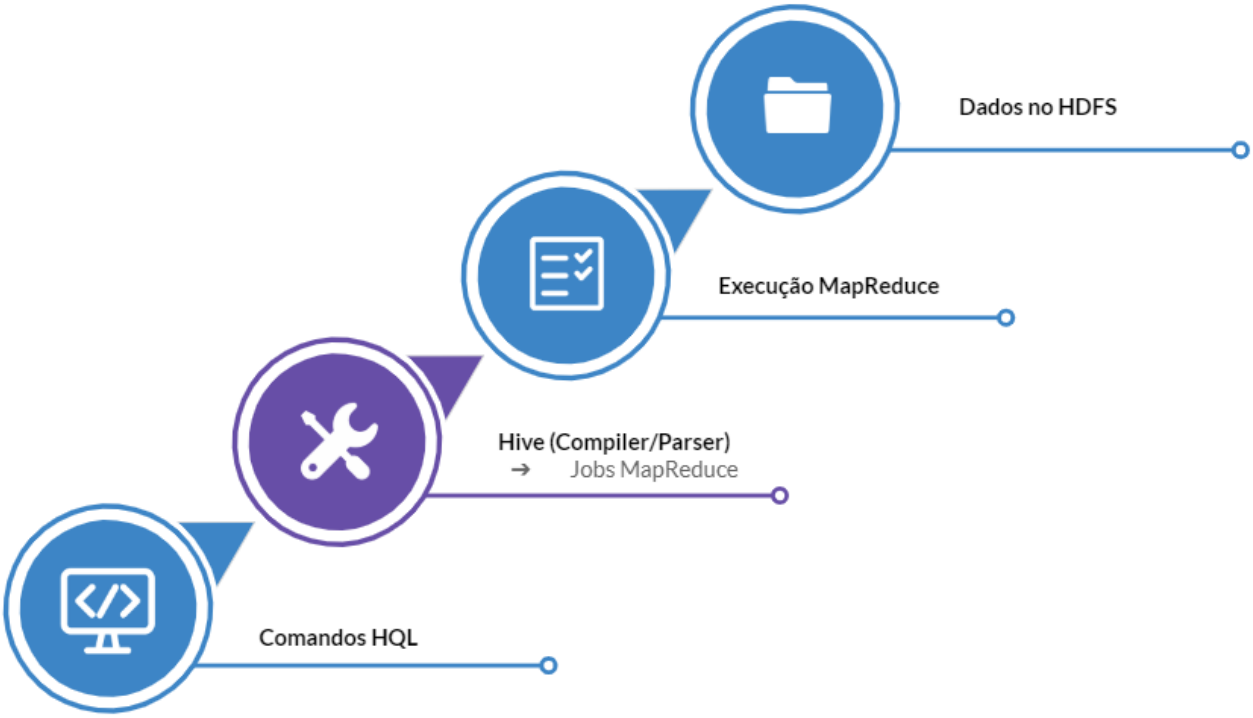
Fonte: White, 2015.

4.2.4. Camada de Acesso aos Dados

Na camada de acesso aos dados, podem ser utilizadas as ferramentas *Hive*, *Sqoop* e *Mahout*. Essa camada tem como principal objetivo facilitar o acesso e manipulação dos dados dentro do *cluster Hadoop* por meio de linguagens mais simples, as quais transformam seus comandos em *Jobs MapReduce*.

Por ser um repositório de dados, o *Hive* está especificado também na camada de armazenamento e pode ser utilizado para realizar a leitura, gravação e o gerenciamento de dados estruturados, utilizando a linguagem *HQL* (*Hive Query Language*) que facilita o acesso aos dados distribuídos por tratar-se de uma linguagem fácil e muito semelhante à linguagem *SQL* (*Structured Query Language*), transformando as sentenças *HQL* em *Jobs MapReduce*. O *Hive* organiza os dados em tabelas e provê um meio para anexar a estrutura aos dados armazenados no *HDFS*, já os metadados são armazenados em um banco de dados chamado *metastore*. O *Hive* possui escalabilidade consistente com o *Hadoop*, permitindo o processamento de dados em larga escala. Os comandos *HQL* são enviados ao *Hive* por meio de uma *interface JDBC* (*Java Database Connectivity*). O *Hive* então compila e analisa as instruções *HQL* em tarefas *MapReduce* para execução no *cluster Hadoop*. O fluxo de execução dos comandos *HQL* é apresentado na Figura 11.

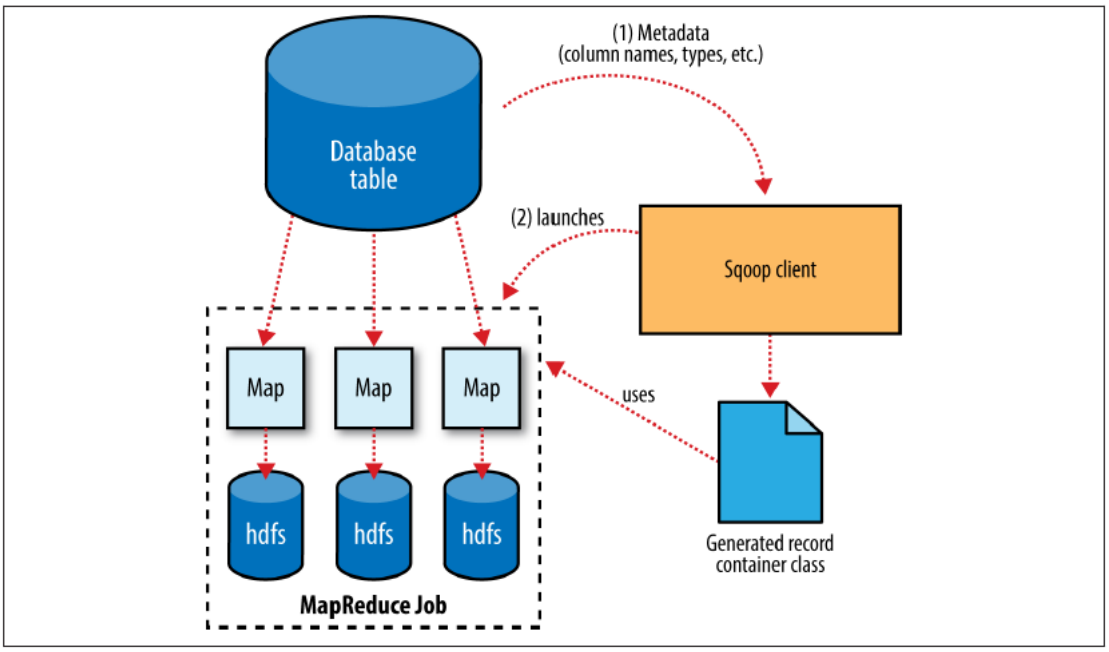
Figura 11: Fluxo de Execução dos Comandos HQL.



Fonte: Autor, 2019.

Para interagir com dados que estejam em repositórios de armazenamento fora do *HDFS*, os programas *MapReduce* necessitam usar *APIs* externas. Nesse caso, o *Sqoop* pode ser aplicado para exportar dados que já foram processados e tratados do *Hadoop* para *datastores* estruturados externos que podem servir de base para relatórios e *dashboards*. Também é possível usar o *Sqoop* para mover dados de um banco de dados externo para o *HBase*. O *Sqoop* importa uma tabela de um banco de dados executando um *Job MapReduce* que extrai linhas da tabela e grava os registros no *HDFS*. Assim como no *Hive*, a interação entre o *Sqoop* e o banco de dados externo ocorre por meio de *interface JDBC*. A tabela a ser importada é examinada, recuperando-se as colunas e os tipos de dados *SQL*, os quais são mapeados para tipos de dados *Java* para manter os valores dos campos nos aplicativos *MapReduce*. O gerador de código do *Sqoop* usará essas informações para criar uma classe específica da tabela para manter um registro extraído da tabela (White, 2015). A Figura 12 apresenta esse processo de importação.

Figura 12: Processo de Importação do *Sqoop*.



Fonte: White, 2015.

Finalizando esta camada propomos a utilização do *Mahout* para implementação de algoritmos de aprendizado de máquina em larga escala, através de bibliotecas *java*, para realizar tarefas de classificação, *clustering*, *data mining* e busca por padrões. Por se tratar de um *framework* recente (versão 0.14 até o presente momento), ainda existem poucas bibliotecas implementadas, tendo como foco principal a recomendação. As principais bibliotecas e os algoritmos implementados em cada categoria estão apresentados na Tabela 5, sendo apenas *Naive Bayes* e *K-Means* com suporte a execução paralela na versão 0.14.

Tabela 4: Lista de Algoritmos de Aprendizado de Máquina Implementados no *Apache Mahout*.

Categoria	Algoritmo
Collaborative Filtering	Item-based Collaborative Filtering
	Matrix Factorization with Alternating Least Squares
	Matrix Factorization with Alternating Least Squares on Implicit Feedback
Classification	Naive Bayes
	Complementary Naive Bayes

	Random Forest
Clustering	Canopy Clustering
	k-Means Clustering
	Fuzzy k-Means
	Streaming k-Means
	Spectral Clustering
Dimensionality Reduction	Lanczos Algorithm
	Stochastic SVD
	Principal Component Analysis
Topic Models	Latent Dirichlet Allocation
Miscellaneous	Frequent Pattern Matching
	RowSimilarityJob
	ConcatMatrices
	Colocations

Fonte: Adaptado de Apache, 2020.

4.2.5. Camada de Análise de Dados

A camada de análise de dados é a parte da arquitetura que implementa os principais processos necessários para gerar conhecimento, em forma de relatórios ou previsões. É possível utilizar monitoramentos, sistemas de recomendação, *dashboards*, etc., objetivando fornecer insumos para a tomada de decisão com relação a veracidade de notícias. Esta camada aplica técnicas descritivas e preditivas, para fornecer características do objeto de estudo, aplicando-se tabelas e gráficos para representar seus resultados, bem como a classificação e previsão do problema analisado. Para este último, é necessário que os modelos sejam estimados e treinados utilizando métodos de aprendizado, aplicados na camada anterior, de maneira contínua com novos dados, periódica ou sob demanda. A escolha das abordagens vai depender dos recursos computacionais disponíveis.

Abaixo são exemplificadas análises que podem ser aplicadas nesta camada:

- Relatórios periódicos.
- Verificação de notícias específicas com base no modelo preditivo.
- Monitoramento de conteúdo.
- Predição de *fake news*.

4.3. Considerações Finais

Este capítulo teve como objetivo demonstrar toda a arquitetura de *big data* proposta no contexto da detecção de notícias falsas em conteúdos digitais. Foram detalhadas as camadas pertencentes à arquitetura, o objetivo de cada camada bem como as ferramentas e/ou tecnologias utilizadas e aplicadas ao domínio escolhido. No próximo capítulo serão apresentados os experimentos realizados bem como os resultados alcançados.

5. EXPERIMENTOS E RESULTADOS

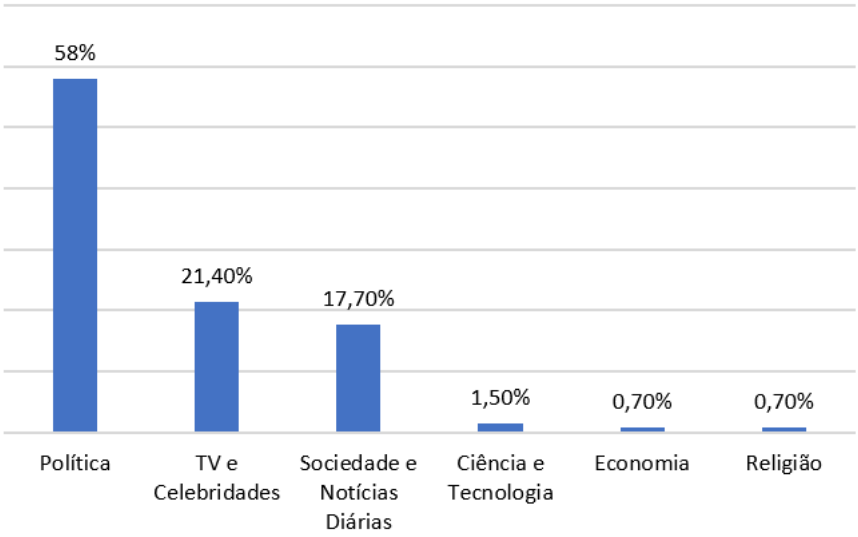
5.1. Considerações Iniciais

Este capítulo consolida a arquitetura proposta, utilizando um conjunto de tecnologias adequados ao contexto de *big data* focado na análise de notícias e apresenta os resultados obtidos a partir dos experimentos realizados.

5.2. Experimentos

Para os experimentos desta pesquisa foram utilizadas três fontes de dados: *Fake.Br Corpus*, desenvolvido por Monteiro (2018); imagens de *fake news* extraídas do *site* do ministério da saúde (Ministerio da Saude, 2020) e notícias extraídas através de portais de notícias e de boatos. O *Fake.Br corpus* é composto de 7200 notícias, sendo 3600 verdadeiras (*true*) e 3600 falsas (*fake*), criadas durante um período de 2 anos, entre 2016 e 2018. As notícias deste *corpus* podem ser divididas nas categorias mostradas no Gráfico 5.

Gráfico 5: Quantidade de Documentos por Categoria no *Fake.Br Corpus.*



Fonte: Adaptado de Monteiro et al., 2018.

Os dados obtidos do ministério da saúde contêm imagens enviadas através do aplicativo *whatsapp* disponibilizado para a população. Foram extraídas 50 notícias falsas e 50 notícias verdadeiras. As imagens foram armazenadas diretamente no *HDFS*. Os

textos das imagens foram extraídos através de algoritmo, utilizando a linguagem de programação *python*, conforme trecho apresentado na Figura 13:

Figura 13: Código para Extração de Textos das Imagens.

```
1 from PIL import Image
2 import pytesseract
3
4
5 for i in range(1,50):
6     nome_arq = 'True/'+ str(i) +'.txt'
7     arq = open(nome_arq, 'w')
8     imagem = 'Imagens_True/' + str(i) + '.jpg'
9     arq.writelines(pytesseract.image_to_string(Image.open(imagem)))
10    arq.close()
11
12 for i in range(1,50):
13    nome_arq = 'Fake/'+ str(i) +'.txt'
14    arq = open(nome_arq, 'w')
15    imagem = 'Imagens_Fake/' + str(i) + '.jpg'
16    arq.writelines(pytesseract.image_to_string(Image.open(imagem)))
17    arq.close()
```

Fonte: Autor, 2019.

O algoritmo apresentado na Figura 13 utiliza dois laços de repetição (linhas 5 a 10 e linhas 12 a 17) para realizar a leitura de cada arquivo com extensão .jpg e extrair os textos destas imagens através do código contido nas linhas 9 e 16 e então armazená-los em arquivos de extensão .txt. O primeiro laço é feito apenas para as imagens verdadeiras (*true*) e o segundo laço para as imagens falsas (*fake*).

Para completar o *corpus* de notícias para os experimentos, foram extraídas 1700 notícias de portais de notícias, atribuídas como verdadeiras e 1700 notícias falsas do *site boatos.org*. Para a extração do portal de notícias, foi utilizado código *PowerShell*, conforme apresentado na Figura 14. Já para a extração do *site boatos.org* foi utilizado algoritmo *python*, apresentado na Figura 15. No código da Figura 14, primeiramente são inicializadas variáveis que definem o *site* a ser utilizado, o número de notícias que serão extraídas do *site* e o diretório onde estas notícias serão armazenadas na máquina local. Posteriormente, o conjunto de notícias é extraído do *site* e armazenado em arquivos com extensão .txt na máquina local. A categoria de cada notícia é armazenada para futura contagem. O código da Figura 15 mostra o método *main*, onde são definidas as variáveis relacionadas ao *site* que será utilizado para o *download* dos dados, a quantidade de páginas que serão visitadas e o código de pesquisa. Neste algoritmo foi utilizado paralelismo através do objeto *Pool* do pacote *multiprocessing*. Após a definição das variáveis, são gerados os *links* que serão visitados pelo algoritmo, considerando a página

inicial e final definidas através das variáveis. Cada *link* é então acessado e seu conteúdo é armazenado dentro de um arquivo com extensão .csv.

Figura 14: Código para Extração de Notícias Verdadeiras.

```
$site = "https://g1.globo.com/"
$num = 1700
$path = "\noticias\g1\"

$ie = new-object -com "InternetExplorer.Application"
$ie.visible = $true
$ie.navigate($site)

while ($ie.Busy -eq $true) { Start-Sleep -seconds 2 }

$noticias = [System.Collections.ArrayList]@()

while ($ie.Document.body.getElementsByClassName("load-more gui-color-primary-bg"))
{
    foreach ($i in ($ie.Document.body.getElementsByClassName("feed-post-link gui-color-primary gui-color-hover") | select href))
    {
        $noticias.Add($i)
    }

    $mais = $ie.document.getElementsByTagName('a') | where-object {$_.innerText -eq 'Veja mais'}
    $mais.click()

    while ($ie.Busy -eq $true) { Start-Sleep -seconds 2 }

    if ($noticias.count -ge $num){
        break
    }
}

foreach ($i in $noticias) {

    $ie.Navigate($i.href)
    # delay
    while ($ie.Busy -eq $true) { Start-Sleep -seconds 2 }

    cd "C:\"
    if (!(Test-Path $path))
    {
        new-item -Name $path -ItemType directory
    }

    $ie.document.body.getElementsByClassName("content-text__container")|
      Select-Object innertext|ForEach-Object {"`t"+$_.innertext *>>("C:\"+$path+([string] $noticias.IndexOf($i))+"g1.txt")}
}
```

Fonte: Autor, 2019.

Figura 15: Trecho de Código para Extração de Notícias Falsas.

```
if __name__ == "__main__":
    site = "boatos.org"
    consulta = '?s=%23boato'
    pag_inicial = 1
    pag_final = 122

    with Pool(5) as p:
        lista_links = p.map(pesquisa_link,
                            range(pag_inicial, pag_final))
        lista_links = [valor for sublista in lista_links for valor in sublista]

        lista_hoaxes = p.map(pesquisa_hoax, lista_links)

        df = pd.DataFrame(lista_hoaxes)
        df = df[df['hoax'].str.len() > 100]
        df.to_csv(site + ".csv")
```

Fonte: Autor, 2019.

As notícias extraídas dos portais de notícias são divididas conforme Tabela 6. As categorias originais foram mantidas, não sendo realizada nenhuma análise ou validação de categorias neste trabalho.

Tabela 5: Quantidade de notícias no *corpus* de notícias verdadeiras.

Categoria	Quantidade	Percentual
Educação	21	1,24%
Meio Ambiente	43	2,53%
Ciência, Tecnologia e Saúde	73	4,29%
Esporte	86	5,06%
Política	109	6,41%
Jornalismo	136	8%
Entretenimento	145	8,53%
Economia	158	9,29%
Mundo	217	12,76%
Brasil	712	41,88%

Fonte: Autor, 2020.

Os textos com menos de 1MB de tamanho foram armazenados em um banco de dados relacional e foram aplicados comandos *HQL* para a realização de algumas checagens de dados e possíveis modificações. As notícias foram então carregadas para o *HDFS* utilizando *Apache Sqoop*. No total, foram utilizadas 10700 notícias, sendo 5350 categorizadas como "*fake*" e 5350 categorizadas como "*true*". São exemplos de notícias, os textos apresentados nas Figuras 16 e 17.

Figura 16: Exemplo de Notícia Falsa no *Corpus*.

```
Joesley chorou ao entrar na cela e está sem comer e tomar banho. O delator Joesley Batista chorou
no momento em que os agentes abriram a cela da carceragem da superintendência da Polícia Federal
em São Paulo para que ele entrasse após se entregar na tarde deste domingo.

Ao ouvir o barulho da grade fechando ele começou a chorar compulsivamente e a tremer. Seus colegas
de cela tentaram o consolar mas ele teve uma crise nervosa e precisou ser medicado. Optou por não
tomar banho (frio) nem se alimentar (marmita). O carcereiro chegou a brincar com ele dizendo que o
bife era de confiança, que era Friboi, mas ele sequer levantou a cabeça.

A Coluna apurou que Joesley e o executivo da JBS Ricardo Saud devem ser transferidos para Brasília
na segunda-feira, 10, em aeronave da Polícia Federal.

Os dois foram presos por determinação do ministro Edson Fachin, do Supremo, atendendo a pedido do
procurador-geral da República, Rodrigo Janot. Eles são acusados de omitir informações no acordo de
delação premiada.

A prisão é temporária, com prazo de cinco dias prorrogáveis por mais cinco.
```

Fonte: Autor, 2019.

Figura 17: Exemplo de Notícia Verdadeira no *Corpus*.

```
A Agência Nacional de Vigilância Sanitária (Anvisa) proibiu a importação, o uso e venda do insumo
farmacêutico ativo valsartana, fabricado pelas empresas Zhejiang Tianyu Pharmaceutical Co. Ltd, da
China, e Hetero Labs Limited, da Índia.

Essa substância é usada na fabricação de medicamentos para tratar hipertensão arterial. A medida
considerou o comunicado de inspeção conduzida pelo European Directorate for the Quality of
Medicines & HealthCare - EDQM, nas empresas.

A inspeção identificou deficiências ligadas a presença da impureza tóxica N-nitrosodimetilamina
(NDMA), o que é classificado como elevado risco sanitário para a saúde pública.

As N-nitrosaminas são compostos que podem ter efeitos cancerígenos. Atualmente, elas estão
presentes em diversos produtos como alimentos, bebidas, medicamentos, cigarros e outros.

A Anvisa não divulgou quais medicamentos ou fabricantes estariam fazendo uso do produto suspenso.

No entanto, foi determinada a suspensão da fabricação, manipulação, distribuição, comercialização
e uso de medicamentos e outros produtos que contenham a valtarsana importada fabricada pelas
empresas citadas.
```

Fonte: Autor, 2019.

Os recursos de *hardware* e *software* utilizados durante os experimentos são detalhados abaixo:

- Recursos de *hardware*:
    - Processador: Intel Xeon X5660 2.80 GHz.
    - Espaço em disco: Tamanho total de 1TB, sendo dividido da seguinte maneira:
        - 200GB para cada máquina virtual do *cluster* (*name-node*, *data-node01*, *data-node02*), totalizando 600GB.
        - 400GB para a máquina real.
    - Memória: 32GB de memória RAM no total, dividida da seguinte forma:
        - 8GB para o *host* real.
        - 24GB para o *cluster*, sendo 8GB para cada VM (*name-node*, *data-note01* e *data-node02*).
- Recursos de *software*:
    - Sistemas Operacionais:
        - Na máquina real: *Linux Mint* 19.
        - Nas máquinas virtuais: *CentOS* 7.6.
    - *Oracle Virtual Box* versão 6.0 para *Linux*.
    - *Hadoop* versão 2.9.2.
    - *Hive* versão 2.3.4.
    - *Sqqop* versão 1.4.7.

- *Mahout* versão 0.13.0.
- Banco de Dados *MariaDB* 10.3.

Para criar um classificador automático de *fake news*, foram realizados testes utilizando aprendizado de máquina, de forma distribuída através dos nós do *cluster Hadoop*. Já que o objetivo é a execução de maneira distribuída, foi escolhido o algoritmo *Naive Bayes* do pacote *Mahout* para a classificação das notícias, uma vez que existem limitações na execução de outros algoritmos utilizados na literatura, como SVM (*Support Vector Machine*), ao longo do *cluster*. Na Figura 18 são apresentados os códigos utilizados para a execução do algoritmo *Naive Bayes* sobre a base de dados. O detalhamento de cada comando é definido através de comentários no código.

Figura 18: Código para Classificação de Notícias.

```
# Conversão dos dados para Sequence:
mahout seqdirectory -i /mahout/fakenews -o /mahout/fakenews/output/seqoutput

# Criação de Vetores TF-IDF:
mahout seq2sparse -i /mahout/fakenews/output/seqoutput
-o /mahout/fakenews/output/sparseoutput

# Separação de Dados para Treinamento e Teste:
mahout split -i /mahout/fakenews/output/sparseoutput/tfidf-vectors
--trainingOutput /mahout/fakenews/nbTrain --testOutput /mahout/fakenews/nbTest
--randomSelectionPct 30 --overwrite --sequenceFiles -xm sequencial

# Construção do Modelo Preditivo:
mahout trainnb -i /mahout/fakenews/nbTrain -li /mahout/fakenews/nbLabels
-o /mahout/fakenews/nbmodel -ow -c

# Teste do Modelo:
mahout testnb -i /mahout/fakenews/nbTest -m /mahout/fakenews/nbmodel
-l /mahout/fakenews/nbLabels -ow -o /mahout/fakenews/nbpredictions -c
```

Fonte: Autor, 2019.

Para a criação dos vetores de números, foi aplicado o método TF-IDF, o qual realiza dois cálculos: no primeiro, referente ao TF (*Term Frequency*), é calculado o número de vezes que uma palavra aparece em um documento dividido pelo número total de palavras no documento; no segundo, relacionado ao IDF (*Inverse Data Frequency*), é calculado o *log* do número de documentos dividido pelo número de documentos que contém a palavra em questão. Este último determina o peso de palavras raras dentro de todos os documentos (Maklin, 2019). Após a criação dos vetores TF-IDF, foi realizada a separação dos dados para treinamento e para teste, considerando 70% dos dados para treinamento e 30% dos dados para teste. Em seguida o modelo preditivo foi criado e testado. As Tabelas 7 e 8

mostram o resumo dos resultados alcançados com o algoritmo e a matriz de confusão, respectivamente.

Tabela 6: Resumo *Naive Bayes*.

Instâncias classificadas corretamente	3133	99,74%
Instâncias classificadas incorretamente	8	0,26%
Total de instâncias classificadas	3141	

Fonte: Autor, 2019.

Tabela 7: Matriz de Confusão.

Classificado como	*Fake*	*True*
Fake	1568	7
True	1	1565

Fonte: Autor, 2019.

Para validar o modelo, foram calculadas métricas tradicionais de acurácia, precisão e *recall*, apresentados na Tabela 9.

Tabela 8: Estatísticas da Classificação.

Acurácia	99,7453%
Precisão	0,9975
Recall	0,9975

Fonte: Autor, 2019.

Podemos observar a obtenção de 99,74% de acurácia no modelo utilizado. Na matriz de confusão para a classificação das notícias podemos observamos bons resultados, uma vez que apenas uma notícia verdadeira foi classificada como falsa e sete notícias falsas foram classificadas como verdadeiras, no conjunto de dados de teste. Obviamente é possível realizar melhorias, já que consideramos a classificação de notícias falsas como verdadeiras mais prejudicial do que o oposto. Houve tentativa de utilização do algoritmo *Random Forest* com o intuito de gerar um comparativo com o algoritmo *Naive Bayes*, porém não foi possível sua execução, pois tal algoritmo foi retirado da versão do *Apache Mahout* utilizada durante os experimentos.

5.3. Considerações Finais

Este capítulo foi importante para a apresentação dos resultados obtidos com a utilização da arquitetura proposta, detalhando os procedimentos utilizados para compor a base de dados de notícias, obtendo-se um conjunto de dados heterogêneo e compondo um *corpus* com 10700 notícias. A classificação foi realizada através da aplicação do método TF-IDF para a estruturação dos documentos e aplicando-se o algoritmo *naive bayes*. Os resultados obtidos foram satisfatórios aos propósitos deste trabalho, pois foi possível compreender o comportamento de uma arquitetura de *big data* para análise de dados com volume considerável e formatos distintos.

6. CONCLUSÕES

Este trabalho teve como objetivo propor uma arquitetura de *big data* que permita o armazenamento, processamento e análise de um grande volume de notícias com formatos variados e que possibilite a implementação de algoritmos de aprendizado de máquina para obtenção de melhores classificadores de notícias. Para tanto, foi realizada uma exposição conceitual de *big data*, incluindo sua definição e características, os desafios decorrentes das peculiaridades do *big data* e os principais paradigmas e tecnologias associados. Com base no enquadramento conceitual e nas arquiteturas existentes, utilizadas para tratar diversos contextos de *big data*, foram identificadas as principais tecnologias envolvidas no processamento de *big data*, com foco especial no ecossistema *Hadoop* e no modelo de programação *MapReduce*, os quais permitem o armazenamento e processamento distribuídos, através da utilização de *clusters* de computadores de baixo custo.

A arquitetura proposta é composta por cinco camadas divididas em três partes: Fontes de Dados, *Big Data* e Análise de Dados. A primeira camada, denominada Fontes de Dados, detalha os tipos e formatos de dados que podem compor uma base de dados de notícias, dentre os quais destacam-se os textos e as imagens. Na segunda camada, denominada Armazenamento de Dados, são determinados os principais repositórios para o adequado armazenamento dos dados, considerando as características do *big data*, destacando-se o sistema de arquivos distribuído (*HDFS*), o qual permite armazenar arquivos grandes, gerados em alta velocidade, rodando em *hardware* "comum". Nesta camada também é possível a utilização de um banco de dados não relacional (*NoSQL*) e de um banco de dados relacional, dependendo do tipo e objetivos de cada dado a ser armazenado. Na terceira camada, denominada Processamento de Dados, é proposta a utilização do modelo de programação *MapReduce* que permite a execução de programas que realizem análises em grandes conjuntos de dados dentro de um *cluster Hadoop* de forma paralela. Ainda nesta camada é sugerida a utilização de um gerenciador de recursos, denominado *Yarn*. Na quarta camada, denominada Acesso aos Dados, sugere-se o uso de ferramentas que permitam um acesso facilitado aos arquivos armazenados no *cluster Hadoop*, compondo esta camada estão as tecnologias *Hive*, *Sqoop* e *Mahout*, sendo a primeira para leitura, gravação e gerenciamento de dados estruturados através de uma linguagem muito semelhante à linguagem *SQL*, facilitando, assim, a elaboração de comandos, o *Sqoop* para exportação de dados entre a plataforma do *big data* e plataformas externas ou entre um banco de dados relacional e o sistema de arquivos distribuído, e o

Mahout para a implementação de algoritmos de aprendizado de máquina. Por fim, na camada de Análise de Dados, propõe-se a implementação de processos necessários para geração de conhecimento, de diversas maneiras, tais como relatórios, gráficos, monitoramentos, painéis, sistemas de recomendação, etc.

Para validar a arquitetura proposta, no contexto da classificação de notícias, foi realizada a implementação de uma infraestrutura com a utilização das tecnologias e métodos que perpassam as cinco camadas. Para a camada de fonte de dados, foi criada uma base de dados de notícias composta por arquivos em formato texto e imagem, totalizando 10700 notícias, sendo 5350 classificadas como verdadeiras (*true*) e 5350 classificadas como falsas (*false*). Estes dados foram armazenados utilizando *HDFS* e *MariaDB*. Para o processamento, foram utilizadas as tecnologias da camada de acesso, ou seja, *Hive*, *Sqoop* e *Mahout*, as quais convertem os comandos criados em *jobs MapReduce* para serem executados no *cluster*. Para criação do modelo preditivo, os dados foram divididos em treinamento e teste, sendo 70% para o primeiro conjunto de dados e 30% para o segundo. Foi então aplicado o algoritmo *Naive Bayes*, disponível no pacote *Mahout*, para testar o modelo criado, obtendo-se uma acurácia de 99,74%.

Com base nos estudos desta pesquisa bem como nos experimentos realizados, conclui-se que é possível a utilização de uma infraestrutura de *big data* que permita a execução de algoritmos de aprendizado de máquina sob um grande volume de dados de maneira distribuída, obtendo-se resultados satisfatórios.

6.1. Contribuições

A principal contribuição deste trabalho consiste na elaboração de uma proposta de arquitetura de *big data* voltada para a identificação de notícias falsas, sendo validada através de experimentação e testes. Tal arquitetura pode ser disponibilizada para a população com o objetivo de verificar a veracidade de notícias, identificar fontes confiáveis e alertar a população para notícias potencialmente falsas. As demais contribuições são resumidas conforme abaixo:

- Estudo sobre as arquiteturas de *big data* utilizadas na literatura e aplicadas a diversos contextos.
- A Arquitetura proposta é escalável e pode facilmente ser estendida a outros contextos que possuam fontes de dados e características semelhantes.
- Composição de uma base de dados de notícias para utilizações futuras.
- Algoritmos para coleta automática de notícias, que podem ser agendados para execução periódica.

Ademais, a divulgação do trabalho por meio de aceitação e publicação de artigo em conferências, onde são apresentadas a proposta e os resultados desta dissertação. A arquitetura conceitual proposta foi submetida e aceita na conferência abaixo detalhada:

— **International Conference on ICT, Society and Human Beings (ICT -2019).**
A Big Data Approach for Fake News Detection: Importance, Tools, and Architecture.

A proposta e os resultados do experimento desta dissertação foram publicados por meio de artigo na conferência abaixo:

— **Encontro Anual de Tecnologia da Informação (EATI-2019).**
A Big Data Architecture for Fake News Detection.

6.2. Trabalhos Futuros

Como trabalhos futuros, espera-se o aprimoramento da proposta com a utilização de Inteligência Artificial (IA) combinando técnicas em várias das camadas da arquitetura a fim de prever comportamentos futuros com precisão crescente, automação de decisões e gerenciamento de desempenho, obtendo-se resultados mais confiáveis e assertivos. Além disso, na camada de Acesso aos Dados, espera-se utilizar outros algoritmos de classificação disponíveis para realizar comparativos com os resultados obtidos neste trabalho, bem como criar um volume maior de notícias, incluindo outros formatos, de maneira a usufruir de todo o potencial da arquitetura no tratamento de *big data*, além de elaborar aplicações para coleta automática de notícias, atualização do modelo preditivo e *interface* para que usuários possam consultar a veracidade de notícias. Também é possível a utilização de tecnologias de processamento *in-memory*, como *Spark*, com o objetivo de

aumentar a velocidade do processamento. Para uma solução completa, é importante a existência de um modelo de segurança em todos os processos.

6.3. Dificuldades Encontradas

Durante o desenvolvimento deste trabalho, algumas dificuldades foram encontradas no que tange as técnicas e tecnologias envolvidas, a captação de trabalhos aplicados ao contexto desta dissertação e a implementação dos algoritmos. No geral, tais obstáculos não impediram a realização e conclusão deste trabalho, porém alguns deles não puderam ser contornados, destacando-se:

- Não foram encontrados trabalhos publicados relacionados com a utilização de uma arquitetura de *big data* no contexto de *fake news*.
- Na fase de implementação, a simulação de vários *clusters* em uma mesma máquina ocasionou a perda de desempenho dos experimentos, uma vez que a solução é indicada para utilização em máquinas distintas.
- O pacote escolhido para compor a solução na fase de implementação dos algoritmos de aprendizado de máquina encontra-se em suas fases iniciais, sendo, portanto, difícil encontrar material de apoio para a elaboração e compreensão dos algoritmos e seus resultados. Além disso, o pacote apresenta limitações na execução distribuída de algoritmos, restringindo as possibilidades de escolha.

REFERÊNCIAS

ABDELOUARIT, K. A.; SBIHI, B.; AKNIN, N. (2017). Towards an Approach Based on Hadoop to Improve and Organize Online Search Results in Big Data Environment. In: International Conference on Communication, Management and Information Technology (ICCMIT).

ALLCOTT, H; GENTZKOW, M. (2017). Social Media and Fake News in the 2016 Election. Journal of Economic Perspectives, 31, 211-236.

ALPAYDIN, E. (2016). Machine Learning The New IA. The MIT Press.

AMARAL, F. (2016). Introdução à Ciência de Dados - Mineração de Dados e Big Data. 1 ed. Alta Books.

APACHE (2019a). Apache Hadoop. Disponível em: <https://hadoop.apache.org>. Acesso em: 25 junho 2019.

APACHE (2019b). HDFS Architecture Guide. Disponível em: <https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html>. Acesso em 25 junho 2019.

APACHE (2019c). MapReduce Tutorial. Disponível em: <https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html>. Acesso em 25 junho 2019.

APACHE (2019d). Mahout For Creating Scalable Performant Machine Learning Applications. Disponível em: <https://mahout.apache.org>. Acesso em: 25 junho 2019.

APACHE (2019e). Welcome to Apache HBase. Disponível em: <https://hbase.apache.org>. Acesso em: 25 junho 2019.

APACHE (2020). Apache Mahout. Disponível em: <https://mapr.com/products/product-overview/apache-mahout>. Acesso em: 03 janeiro 2020.

BERMAN, F.; FOX, G.; HEY, A.J. (2003). Grid Computing: Making the Global Infrastructure a Reality. Wiley.

BORUNDA, M. et al. (2016). Bayesian networks in renewable energy systems: A bibliographical survey. Renewable and Sustainable Energy Reviews, 2016. v. 62, p. 32–45. ISSN 1364-0321.

BRAGA, M. J. (2018). Voto no parecer do Conselho de Comunicação Social nº 1. Disponível em: <https://www25.senado.leg.br/web/atividade/materias/-/materia/133519>. Acesso em: 03 fevereiro 2020.

BREIMAN, L.; FRIEDMAN, J.H.; OLSHEN, R.A.; STONE, C.J. (1984). Classification and Regression Trees. Wadsworth.

BRYANT, R.E.; KATZ, R.H.; LAZOWSKA, E.D. (2008). Big data computing: creating breakthroughs in commerce, science, and society. Comunity Computing Consortium. Disponível em: <https://cra.org/ccc/wp-content/uploads/sites/2/2015/05/Big_Data.pdf>. Acesso em: 25 dezembro 2019.

BURSHTEIN, S. (2017). The True Story on Fake News. Intellectual Property Journal, 29(3), 397-447.

CARILLET, D. (2019), Fake News. Disponível em: <https://mundoeducacao.bol.uol.com.br/curiosidades/fake-news.htm>. Acesso em: 28 jun 2019.

CASTRO, F. F.; BAÍA, D.C.P. (2012). Uma experiência de inclusão digital na Amazônia: o Programa NavegaPará e o horizonte da democracia. Limites e Esperanças. Porto Alegre: FAMECOS.

CHANDARANA, P.; VIJAYALAKSHMI, M. (2014). Big Data analytics frameworks. In International Conference on Circuits, Systems, Communication and Information Technology Applications (CSCITA) p. 430–434. Disponível em: <http://doi.org/10.1109/CSCITA.2014.6839299>.

CHEN, M.; MAO, S.; LIU, Y. (2014). Big Data: A Survey. In Mobile Networks and Applications. 19(2), 171–209. Disponível em: <http://doi.org/10.1007/s11036-013-0489-0>.

CHOWDHURY, G.G. (2005). Natural language processing. Annual Review of Information Science and Technology. v.37, p. 51–89.

COMITÊ GESTOR DE INTERNET NO BRASIL (2019). Pesquisa sobre o uso das Tecnologias de Informação e Comunicação nos domicílios brasileiros - TIC Domicílios 2018. Centro Regional de Estudos para o Desenvolvimento da Sociedade da Informação - Cetic.br. Disponível em: <https://cebrap.org.br/wp-content/uploads/2019/10/tic_dom_2018.pdf>. Acesso em: 19 dez 2019.

COULOURIS, G.; DOLLIMORE, J.; KINDBERG, T.; GORDON, B. (2011). Distributed Systems Concepts and Design. 5 ed. Addison Wesley Longman.

DE SOUZA, L. V.; OLIVEIRA, E. L.; RÊGO, L. P.; COSTA, J. C.; FRANCÊS, C. R. (2008). Digital Inclusion on Amazon Region using Wireless Broadband Networks: NavegaPará Project. In: 1st Workshop on Wireless Broadband Access for Communities and Rural Developing Regions, 2008, Karlstad.

DELOITTE (2014). Value of connectivity Economic and social benefits of expanding internet access. Disponível em: <https://www2.deloitte.com/content/dam/Deloitte/ie/Documents/TechnologyMediaCommunications/2014_uk_tmt_value_of_connectivity_deloitte_ireland.pdf>. Acesso em: 27 dez 2019.

DUFFY, B. (2018). Fake News, Filter Bubbles and Post-Truth are Other People's Problems. Disponível em: <https://www.ipsos.com/en/fake-news-filter-bubbles-and-post-truth-are-other-peoples-problems>. Acesso em: 12 set 2019.

EVANS, D. (2011). The Internet of Things: How the Next Evolution of the Internet is Changing Everything. Cisco Internet Business Solutions Group (IBSG). Disponível em: <https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf>. Acesso em: 25 dez 2019.

GANTZ, J.; REINSEL, D. (2011). Extracting Value from Chaos, IDC's Digital Universe Study. EMC Corporation.

GANTZ, J.; REINSEL, D. (2012). The Digital Universe In 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East. EMC Corporation. Disponível em: <http://www.emc.com/collateral/analyst-reports/idc-the-digital-universe-in-2020.pdf>. Acesso em: 20 dez 2019.

GARTNER. (2019) Big Data defintion. Disponível em: <http://www.gartner.com/it-glossary/big-data/>. Acesso em: 18 dezembro 2019.

HAINDL, M.; SOMOL, P.; VERVERIDIS, D.; KOTROPOULOS, C. (2006). Feature selection based on mutual correlation. In: Martínez-Trinidad, J.F., Carrasco Ochoa, J.A., Kittler, J. (eds.) Progress in Pattern Recognition, Image Analysis and Applications. Lecture Notes in Computer Science, v. 4225, p. 569–577. Springer, Berlin, Heidelberg.

HAN, F. (2010). How the Brain Saves Energy: The Neural Thermostat. Disponível em: <http://www.yalescientific.org/2010/09/how-the-brain-saves-energy-the-neural-thermostat>. Acesso em: 25 set 2019.

HAN, J.; KAMBER, M.; PEI, J. (2011). Data Mining. Concepts and Techniques, 3 ed. The Morgan Kaufmann Series in Data Management Systems.

KUMAR, S.; SINGH, M. (2019). Big data analytics for healthcare industry: impact, applications, and tools. Big Data Mining and Analytics, v. 2, n. 1, p. 48-57.
KRISHNAN, K. (2013). Data Warehousing in the Age of Big Data. 1 ed. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.

LEE, K.K.Y. et al. (2013). Alternatives to relational database: Comparison of NoSQL and XML approaches for clinical data storage. In Computer Methods and Programs in Biomedicine. v. 110, n. 1, p. 99–109.

LINLI, W.; ZHANGYI, S.; XIANG, F. (2016). Implementation of massive data processing architecture for electric enterprise groups. IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS).

LIU, Z.; YANG, P.; ZHANG, L. (2013). A Sketch of Big Data Technologies. In 2013 Seventh International Conference on Internet Computing for Engineering and Science (ICICSE) p. 26–29. Disponível em: <http://doi.org/10.1109/ICICSE.2013.13>.

MACEDO, H.R.; CARVALHO, A.X.Y. Aumento da penetração do serviço de acesso à internet em banda larga e seu possível impacto econômico: análise através de sistema de equações simultâneas de oferta e demanda. Rio de Janeiro: Ipea, 2010. Disponível em: <http://goo.gl/noph8x>.

MACHADO, A.L. (2017). Administração do Big Data. 1 ed. Senac, São Paulo.

MAKLIN, C. (2019). TF IDF | TFIDF Python Example. Disponível em: <https://towardsdatascience.com/natural-language-processing-feature-engineering-using-tf-idf-e8b9d00e7e76>. Acesso em: 07 jan 2020.

MINISTERIO DA SAUDE (2020). Saúde sem Fake News. Disponível em: <http://saude.gov.br/fakenews>. Acesso em: 03 jan 2020.

MITCHELL, M.T. (1997). Machine Learning. 1 ed. McGraw-Hill Science.

MONTEIRO, R.A. et al. (2018). Contributions to the Study of Fake News in Portuguese: New Corpus and Automatic Detection Results. In Proceedings of the 13th International Conference. Canela, Brazil, p. 24-26.

MYSORE, D.; KHUPAT, S.; JAIN, S. (2014). Entendendo as camadas de arquitetura de uma solução de big data. Disponível em: <https://www.ibm.com/developerworks/br/library/bd-archpatterns3/index.html>. Acesso em: 11 fev 2020.

OLADIPUPO, T. (2010). Types of Machine Learning Algorithms. New Advances in Machine Learning. Disponível em: <http://cdn.intechweb.org/pdfs/10694.pdf>. Acesso em: 25 dez 2020.

PÉREZ-ROSAS, V. et al. (2018). Automatic Detection of Fake News. In Proceedings of the 27th International Conference on Computational Linguistics. Santa Fe, USA, p. 3391–3401.

PESENSON, M.Z.; PESENSON, I.Z.; MCCOLLUM, B. (2010). The data big bang and the expanding digital universe: high-dimensional, complex and massive data sets in an inflationary epoch. Adv. Astron. p. 1–16. Disponível em: <http://dx.doi.org/10.1155/2010/350891>.

QIANG, C.Z.W.; ROSSOTTO, C.M.; KIMURA, K. (2009). Information and communications for development 2009: extending reach and increasing impact. Washington: WBD. Disponível em: <http://goo.gl/RDQUHy>.

QIU, J.; WU, Q.; DING, G.; XU, Y.; FENG, S. (2016). A survey of machine learning for big data processing. EURASIP Journal on Advances in Signal Processing. p. 67. Disponível em: <https://doi.org/10.1186/s13634-016-0355-x>.

REVATHY, R.; BALAMURALI, S.; LAWRANCE, R. (2019). Classifying Agricultural Crop Pest Data Using Hadoop MapReduce Based C5.0 Algorithm. Journal of Cyber Security and Mobility, v. 8, n. 3, p. 393-408.

RUBIN, V.L.; CHEN, Y.; CONROY, N.J. (2015). Deception detection for news: Three types of fakes. Proceedings of the Association for Information Science and Technology 52(1), 14.

SAGIROGLU, S.; SINANC, D. (2013). Big data: A review. In 2013 International Conference on Collaboration Technologies and Systems (CTS), p. 42–47. Disponível em: <http://doi.org/10.1109/CTS.2013.6567202>.

SCHNEIDER, R.D. (2012). Hadoop For Dummies, Special Edition. John Wiley & Sons, Canada.

SCHUTT, R.; O'NEIL, C. (2013). Doing Data Science: Straight Talk from the Frontline. 1 ed. O'Reilly Media, Inc.

SECTETPA (2019). Programa de Inclusão Digital NAVEGAPARÁ. Belém: Secretaria de Estado de Ciência, Tecnologia e Educação Profissional e Tecnológica. s. d. Disponível em: <http://www.sectet.pa.gov.br/audiovisual/basic-page/navegapar%C3%A1>. Acesso em: 19 dez 2019.

SHU, K. et al (2017). Fake News Detection on Social Media: A Data Mining Perspective. In ACM SIGKDD Explorations Newsletter, v. 19, n. 1, p. 22-36.

SINHA, S. (2019). Hadoop Ecosystem: Hadoop Tools for Crunching Big Data. Disponível em: <https://www.edureka.co/blog/hadoop-ecosystem>. Acesso em: 25 dez 2019.

SOUZA, F. (2018). É como usar drogas: Por que as pessoas acreditam e compartilham notícias falsas. Disponível em: <https://www.bbc.com/portuguese/brasil-45767478>. Acesso em: 29 jan 2020.

TANENBAUM, A.S. (2016). Distributed Systems: Principles and Paradigms. 2 ed. Pearson.

WANG, Y.; KUNG, L.; BYRD, T.A. (2016). Big data analytics: understanding its capabilities and potential benefits for healthcare organizations. Technol. Forecast. Soc. Chang. p. 1–11. Disponível em: <http://dx.doi.org/10.1016/j.techfore.2015.12.019>.

WHITE, T. (2015). Hadoop: The Definitive Guide. 4 ed. O'Reilly.

YUVARAJ, N.; SRIPREETHAA, K.R. (2017). Diabetes prediction in healthcare systems using machine learning algorithms on Hadoop cluster. Clust. Comput. v. 22, p. 1–9.

ZIKOPOULOS, P.; EATON, C. (2011). Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data. 1 ed. McGraw-Hill Osborne Media.