



Universidade Estadual de Campinas
Instituto de Computação



Thales Eduardo Nazatto

Construção de um *Pipeline* de *Machine Learning*
autônomo com métricas de *Fairness*

CAMPINAS
2022

Thales Eduardo Nazatto

Construção de um *Pipeline* de *Machine Learning* autônomo com
métricas de *Fairness*

Dissertação apresentada ao Instituto de
Computação da Universidade Estadual de
Campinas como parte dos requisitos para a
obtenção do título de Mestre em Ciência da
Computação.

Orientadora: Profa. Dra. Cecília Mary Fischer Rubira
Coorientador: Prof. Dr. Leonardo Montecchi

Este exemplar corresponde à versão da
Dissertação entregue à banca antes da
defesa.

CAMPINAS
2022

Na versão final, esta página será substituída pela ficha catalográfica.

De acordo com o padrão da CCPG: “Quando se tratar de Teses e Dissertações financiadas por agências de fomento, os beneficiados deverão fazer referência ao apoio recebido e inserir esta informação na ficha catalográfica, além do nome da agência, o número do processo pelo qual recebeu o auxílio.”

e

“caso a tese de doutorado seja feita em Cotutela, será necessário informar na ficha catalográfica o fato, a Universidade conveniente, o país e o nome do orientador.”

Na versão final, esta página será substituída por outra informando a composição da banca e que a ata de defesa está arquivada pela Unicamp.

Resumo

Esta dissertação de Mestrado possui o objetivo de contribuir com o tema de Inteligência Artificial (IA) do ponto de vista da Engenharia de Software, visto que o uso de IA envolvendo grandes volumes de dados vem crescendo conforme nossa sociedade migra processos manuais de trabalho para soluções digitais e necessita de tomadas de decisão mais rápidas e assertivas. Entretanto, as métricas usadas inicialmente para definir a eficácia de um algoritmo se mostraram limitadas devido a barreiras éticas e legais, resultando em vieses que refletem a sociedade de maneira que não era esperada pelos desenvolvedores da solução. Desta forma, o uso de novas métricas, como métricas de Fairness, e proveniência de dados pode ajudar neste problema. Desenvolvendo um pipeline autônomo utilizando a arquitetura MAPE-K, documentando métodos e ferramentas, e realizando estudos de caso, é esperado que este problema seja controlado e facilite o desenvolvimento de softwares com o uso responsável de dados.

Abstract

This Master's Degree dissertation aims to contribute to the theme of Artificial Intelligence (AI) from the Software Engineering point of view, since the use of AI involving large volumes of data has been growing as our society migrates manual work processes to digital solutions and needs faster and more assertive decision making. However, the metrics used initially to define the effectiveness of an algorithm proved to be limited due to ethical and legal barriers, resulting in biases that reflect society in a way that was not expected by the solution developers. Thus, the use of new metrics, like Fairness metrics, and data provenance can help with this problem. By developing an autonomous pipeline using MAPE-K architecture, documenting methods and tools, and making case studies, it is expected that this problem will be controlled and facilitate software development with the responsible use of data.

Sumário

Resumo	5
Abstract	6
1 Introdução	8
2 Conceitos Relacionados	10
2.1 Ciência de Dados e Engenharia de Dados	10
2.2 Machine Learning	10
2.2.1 Métricas de Avaliação	12
2.2.2 Algoritmos	12
2.3 Fairness em Machine Learning	12
2.3.1 Métricas de Fairness	12
2.4 Algoritmos para redução de vieses	12
2.5 AI Explainability	12
2.6 Engenharia de Software	14
2.6.1 Engenharia de Software para Aplicações de IA	14
2.7 Proveniência de Dados	14
2.8 Arquitetura de Software	14
2.8.1 Arquitetura MAPE-K	14
2.8.2 Arquitetura Pipe-and-Filter	15
3 Metodologia	16
3.1 Detalhamento do processo e diagramas elaborados	16
3.2 Arquitetura do código	16
3.3 Experimentos e limitações	16
4 Resultados e Discussões	17
5 Conclusões	18

Capítulo 1

Introdução

Técnicas de Inteligência Artificial e Aprendizado de Máquina já são utilizadas há bastante tempo no ramo da Computação. Ramos como robótica e jogos são grandes exemplos, dada a necessidade nos mesmos de automatizar comportamentos que seriam tidos como triviais para um ser humano. Entretanto, nos últimos anos ocorreu um crescimento no uso dessas tecnologias em aplicações tradicionais, com previsão de US\$ 57 bilhões em investimentos em 2021, 480% maior em relação a 2017 [5]. No Brasil, o número de empresas de IA aumentou de 120 em 2018 para 206 em 2020 [6].

Isso se mostra possível devido a grande quantidade de dados processada diariamente pelas empresas, que coletam estatísticas toda vez que um usuário acessa suas aplicações. Com esses dados, podem traçar diferentes perfis e usar soluções de IA para ter tomadas de decisão mais assertivas com o objetivo de melhorar a experiência de usuário e corrigir problemas. Porém, muitas dessas soluções foram projetadas sem pensar em governança de dados como requisito de projeto, e se mostram ineficientes quando ela é tratada em consideração.

Governança de dados é um tema que entrou em evidência recentemente em países como o Brasil: Iniciativas como a LGPD - Lei Geral de Proteção de Dados [7], de 14 de agosto de 2018, mostram como as aplicações e seus dados possuem cada vez mais influência na sociedade moderna. E nesse ponto muitas aplicações de IA falham: muitas implementações foram implementadas como *black boxes*, onde o determinante para estabelecer a confiança no modelo implementado é sua entrada e sua saída.

Um outro efeito colateral dessa estratégia é a exposição de vieses que, embora sejam vistos como não-intencionais pelos desenvolvedores por ter a possibilidade de ser um *outlier* no modelo treinado, refletem preconceitos escancarados da sociedade atual. Uma entrada de dados enviesada resulta em um algoritmo que realiza discriminações em sua classificação [10], e uma vez que as métricas utilizadas para medir a qualidade de um modelo *black box* são geralmente baseadas em acurácia, precisão e recall, discriminações não são facilmente percebidas por tais métricas. Ao mesmo tempo, um modelo *black box* pode ter uma alta dependência de poucos dados, determinando problemas de acoplamento. Para resolver este problema, é possível que a criação de um novo modelo seja uma melhor opção que realizar a correção em apenas parte dele.

Devido a esses tipos de problemas, o termo *Explainable AI* (XAI) ganha força para envolver o desenvolvimento de uma IA que seja acurada e simultaneamente transparente.

Como IA possui diversos tipos de métodos diferentes para enquadrar diversos tipos de dados, o mesmo acaba se aplicando em Explainable AI, podendo enquadrar em diversos tipos de dados [19], ou dados específicos como imagens [13] e tabelas [18]. Como o objetivo em XAI é fazer com que os resultados alcançados pela solução de IA sejam compreendidos por humanos, é possível considerar este fato como requisito no design de uma solução de IA, fazendo com que a mesma seja reusável e testável.

No mesmo tema, é possível estabelecer métricas para determinar o quão o modelo está preparado para dados sensíveis [9], termo que é conhecido como *Fairness*. Com a evolução das pesquisas na comunidade acadêmica, foram descobertos algoritmos para redução dos vieses presentes nos conjuntos de dados, como *Reweighting* [11], *Adversarial Debiasing* [20] e *Reject Option Classification* [12]. Por consequência, ocorre melhora nas métricas em questão, mas pode desfavorecer métricas que já são amplamente utilizadas como garantia de um bom modelo desenvolvido com técnicas de Aprendizado de Máquina.

O objetivo desta dissertação de mestrado é desenvolver uma estrutura de *Pipeline* para *Machine Learning* que seja completamente autônoma, por três fatores principais:

- Facilitar a criação de modelos justos e confiáveis com a automatização da escolha dos algoritmos, cuja complexidade aumenta com a escolha dos algoritmos a serem utilizados e suas execuções nas etapas corretas do processo, onde eles foram escolhidos para atuar.
- Estabelecer um balanceamento entre métricas para avaliar bons modelos com métricas para avaliar modelos justos.
- Considerar proveniência de dados como requisito no design de uma solução de IA, e como uma alternativa a XAI através da utilização de metadados.

Para isso, além do desenvolvimento do *Pipeline* em questão, será utilizada a arquitetura MAPE-K [4] para analisar uma base de conhecimento e prover o melhor pipeline seguindo regras pré-determinadas.

O restante da dissertação se organizará da seguinte forma: o Capítulo 2 descreve os conceitos que irão ser abordados neste projeto; o Capítulo 3 mostra a metodologia e detalhamento do processo de desenvolvimento; o Capítulo 4 discute os resultados obtidos e, finalmente, o Capítulo 5 estabelece as conclusões, considerações finais e sugestões de trabalhos futuros e evoluções.

Capítulo 2

Conceitos Relacionados

2.1 Ciência de Dados e Engenharia de Dados

2.2 Machine Learning

Aprendizado de Máquina (*Machine Learning*, em inglês) pode ser definido como “a prática de usar algoritmos para coletar dados, aprender com eles, e então fazer uma determinação ou predição sobre alguma coisa no mundo. Então em vez de implementar as rotinas de software manualmente, com um gama específica de instruções para completar uma tarefa em particular, a máquina é ‘treinada’ usando uma quantidade grande de dados e algoritmos que dão a ela a habilidade de aprender como executar a tarefa” [1]. Com isso, o computador consegue a habilidade de realizar determinado cálculo ou tarefa sem que necessite de programação adicional ou interferência humana para isso.

O *Machine Learning* é fortemente relacionado com a Estatística, uma vez que seus métodos e parte de seus algoritmos, como regressões, tiveram como base modelos estatísticos e a análise de seus dados. As tarefas de aprendizado podem ser classificadas em três categorias básicas [3] [2]:

- **Aprendizado supervisionado:** O treinamento é realizado por meio de exemplos rotulados, como uma entrada na qual a saída desejada é conhecida. Através de métodos como classificação, regressão e *gradient boosting*, o aprendizado supervisionado utiliza padrões para prever os valores de rótulos em dados não-rotulados adicionais. O aprendizado supervisionado é comumente empregado em aplicações nas quais dados históricos preveem eventos futuros prováveis.
- **Aprendizado não-supervisionado:** É utilizado em dados que não possuem rótulos históricos. A “resposta certa” não é informada ao sistema, o algoritmo deve descobrir o que está sendo mostrado. O objetivo é explorar os dados e encontrar alguma estrutura dentro deles. Técnicas populares incluem mapas auto-organizáveis, mapeamento por proximidade, agrupamento *k-means* e decomposição em valores singulares. Esses algoritmos também são utilizados para segmentar tópicos de texto, recomendar itens e identificar pontos discrepantes nos dados.

- **Aprendizado por reforço:** O algoritmo descobre através de testes do tipo “tentativa e erro” quais ações rendem as maiores recompensas. Este tipo de aprendizado possui três componentes principais: o agente (o aprendiz ou tomador de decisão), o ambiente (tudo com que o agente interage) e ações (o que o agente pode fazer). O objetivo é que o agente escolha ações que maximizem a recompensa esperada em um período de tempo determinado. O agente atingirá o objetivo muito mais rápido se seguir uma boa política, então o foco do aprendizado por reforço é descobrir a melhor política.

O termo *Machine Learning* se tornou muito mais evidente com a possibilidade da implementação do *Deep Learning*, que é uma técnica que utiliza Redes Neurais Artificiais para atingir seus resultados. Redes Neurais Artificiais são modelos computacionais inspirados no sistema nervoso do cérebro, onde temos neurônios divididos em camadas e conectados entre si, podendo ser abstraído conforme ilustração na Figura 2.1. Dependendo da tarefa a ser realizada, cada neurônio atribui um peso para os dados que entram e a saída final é determinada pelo total desses pesos [1]. As redes neurais utilizadas em *Deep Learning* possuem, ao menos, duas camadas de neurônios entre a camada que recebe os dados de entrada e a camada final que faz o tratamento final dos dados de saída.

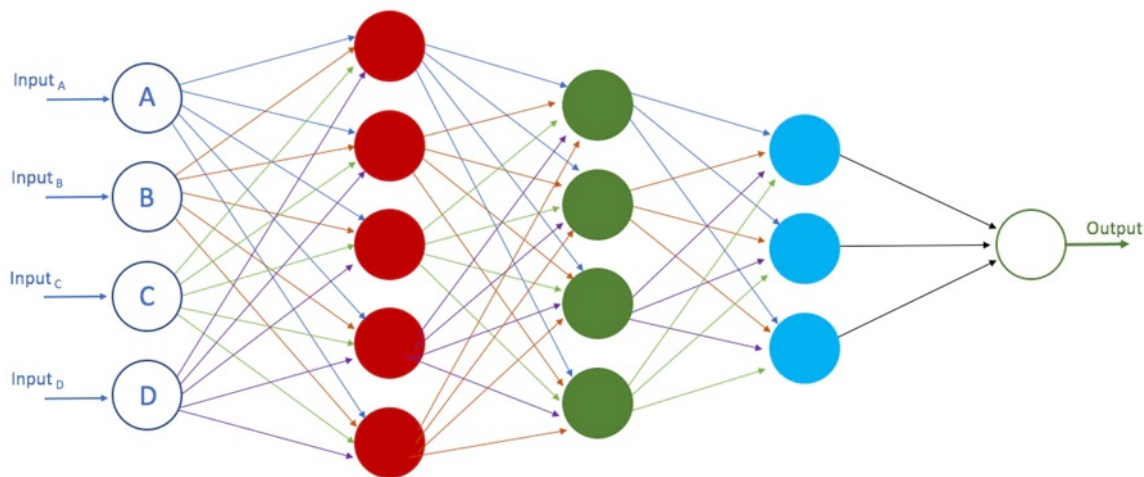


Figura 2.1: Exemplo de uma rede neural utilizada em *Deep Learning*.

Com a evolução da computação, o treino de uma tarefa passou a ser cada vez mais viável, uma vez que a execução de algoritmos de *Machine Learning* é computacionalmente muito custosa, especialmente quando redes neurais são utilizadas. E sua viabilidade é acompanhada de efetividade: Como exemplo, reconhecimento de imagens por máquinas treinadas através de deep learning em alguns cenários possuem uma taxa de acerto maior que a de humanos [1].

2.2.1 Métricas de Avaliação

2.2.2 Algoritmos

2.3 Fairness em Machine Learning

2.3.1 Métricas de Fairness

2.4 Algoritmos para redução de vieses

2.5 AI Explainability

AI Explainability é um conceito em IA que propõe a criação de um conjunto de técnicas de Aprendizado de Máquina (ou Machine Learning/ML) que produz modelos mais explicáveis, mantendo qualidade em suas métricas, e permite que os humanos entendam, confiem e gerenciem aplicações baseadas em IA. XAI também absorve conceitos das Ciências Sociais e considera a psicologia da explicação [8]. Um algoritmo de ML explicável precisa não apenas mostrar tomadas de decisão, mas também mostrar o processo que o levou ao tomar tal decisão, de modo que seja compreensível e transparente para humanos.

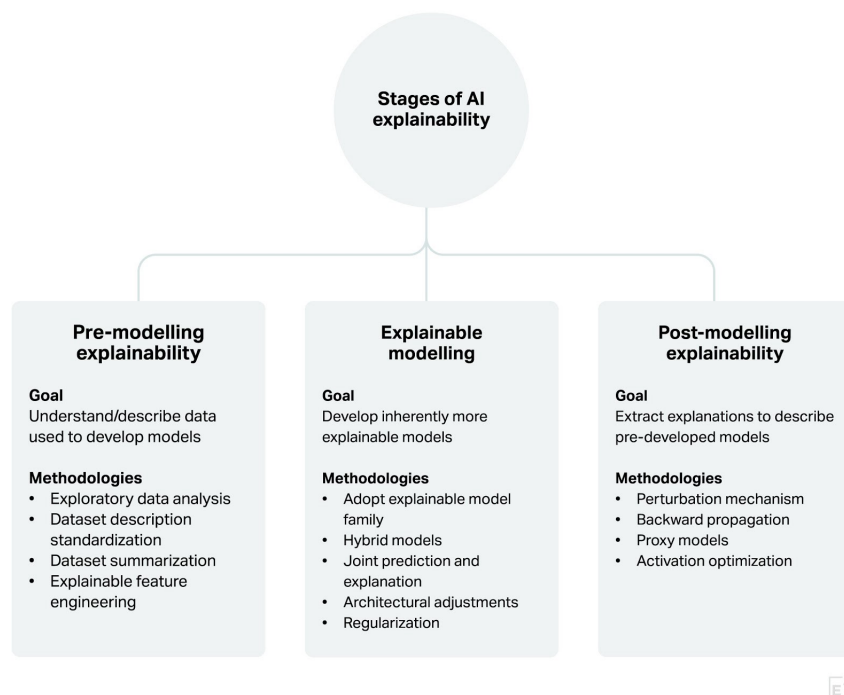


Figura 2.2: Fases de um processo baseado em IA explicável.

Conforme ilustrado na Figura 2.2, o um processo baseado em IA explicável pode ser classificado em 3 fases:

- **Pre-Modelling Explainability:** Esta fase se caracteriza por obter explicações no conjunto de dados usado para treino e validação, tendo como principal motivação o

fato do comportamento de um modelo depender muito dos dados que o alimentam. Análises e visualizações dos dados, ou mesmo através de documentações do conjunto de dados se enquadram neste estágio. [17]

- ***Explainable Modelling***: Esta fase se caracteriza por obter explicações por modelos considerados transparentes (ver adiante). Ao adotar modelos deste tipo, ele já pode ser considerado como explicável por um humano. Também é possível obter a explicação através de regularizadores, ou adotar abordagens onde explicação e resultado sejam obtidos simultaneamente, sejam via junção de dados ou sendo intrínseco a arquitetura do modelo. [15]
- ***Post-Modelling Explainability***: Também chamada de *Post-hoc Explainability* e onde a maioria dos trabalhos disponíveis baseados em XAI tenta focar, esta fase se dedica a explicar modelos desenvolvidos anteriormente, ou modelos que não são considerados como transparentes. Pode-se explicar um modelo por métodos como árvores de decisão e estimativas como valores de Shapley e propagação inversa. [16]

Entretanto, estas 3 fases não necessariamente são executadas de maneira sequencial: Por ser realizada no conjunto de dados e não no modelo, a fase de *Pre-modelling Explainability* pode ser opcional, e a aplicação das fases de *Explainable Modelling* e *Post-Modelling Explainability* e seus métodos dependem exclusivamente se o modelo é classificado como transparente ou não, e tal critério é determinado pela adoção de uma (ou mais) das seguintes características [8]:

- **Simulabilidade**: Denota a capacidade de um modelo de ser simulado ou pensado estritamente por um humano. A complexidade assume um lugar dominante nesta classe: Sistemas com uma grande quantidade de regras, por mais simples que essas sejam, já não podem ser classificados como tal. O modelo que se adequa a essa característica precisa ser autocontido o suficiente para que um ser humano pense e raciocine sobre ele como um todo.
- **Decomponibilidade**: Também considerado como inteligibilidade, significa a capacidade de explicar cada uma das partes de um modelo. Se o modelo não for simples o suficiente, ele precisa ser divisível em várias pequenas partes e cada parte do modelo deve ser compreensível por um ser humano, sem a necessidade de ferramentas adicionais.
- **Transparência do algoritmo**: Trata-se da habilidade do usuário de entender o processo seguido pelo modelo para produzir qualquer saída dada a partir de seus dados de entrada. Colocando de outra forma, um modelo linear é considerado transparente porque sua superfície de erro pode ser entendida e fundamentada, permitindo ao usuário entender como o modelo irá agir em cada situação que pode enfrentar.

2.6 Engenharia de Software

2.6.1 Engenharia de Software para Aplicações de IA

2.7 Proveniência de Dados

2.8 Arquitetura de Software

2.8.1 Arquitetura MAPE-K

Em 2001, Paul Horn introduziu o conceito de Computação Autônoma como alternativa a solução para a crescente complexidade dos sistemas da época, onde previa-se que os mesmos se tornariam muito grandes e complexos até mesmo para os profissionais mais qualificados configurarem e realizarem manutenção. Tal conceito qualifica sistemas de computação que podem se autogerenciar com relação aos objetivos de alto nível dados pelos administradores e é derivado da biologia, dado a grande variedade e hierarquia de sistemas autônomos presentes na natureza e na sociedade [14].

Em um ambiente autônomo e autogerenciado, os componentes de sistema podem incorporar como funcionalidade um *loop* de controle. Embora estes *loops* sejam divididos nos mesmos procedimentos, é possível categorizá-los em 4 categorias principais. Essas categorias são consideradas atributos dos componentes do sistema e são definidas como [4]:

- **Auto-configuração:** Pode se adaptar dinamicamente mudanças no ambiente. Um componente autoconfigurável realiza esta adaptação usando políticas fornecidas pelo profissional. Tais mudanças podem incluir a implantação de novos componentes ou a remoção dos existentes, ou mudanças drásticas nas características do sistema. A adaptação dinâmica ajuda a garantir força e produtividade contínuas da infraestrutura, resultando em crescimento e flexibilidade dos negócios.
- **Auto-cura:** Pode descobrir, diagnosticar e reagir a interrupções. Um componente auto-curável pode detectar falhas no sistema e iniciar ações corretivas baseadas em políticas sem interromper o ambiente. A ação corretiva pode envolver um produto alterando seu próprio estado ou efetuando mudanças em outros componentes do ambiente. Com isso, o sistema se torna mais resiliente porque as operações cotidianas possuem menos probabilidade de falhar.
- **Auto-otimização:** Pode monitorar e ajustar recursos automaticamente. Um componente auto-otimizável pode se ajustar para atender às necessidades do usuário. As ações de ajuste podem significar realocar recursos para melhorar a utilização geral, como em resposta a cargas de trabalho que mudam dinamicamente, ou garantir que processamentos possam ser concluídos em tempo hábil. A auto-otimização ajuda a fornecer um alto padrão de serviço para quem vai utilizar o sistema. Sem funções de auto-otimização, não há uma maneira fácil de re-escalonar os recursos de infraestrutura quando um aplicativo não os usa totalmente.

- **Auto-proteção:** Pode antecipar, detectar, identificar e proteger contra ameaças de qualquer lugar. Um componente de autoproteção pode detectar comportamentos hostis à medida que ocorrem e tomar ações corretivas para se tornarem menos vulneráveis. Os comportamentos hostis podem incluir acesso e uso não autorizados, infecção e proliferação de vírus e ataques de negação de serviço. Os recursos de autoproteção permitem que as empresas apliquem consistentemente políticas de segurança e privacidade.

–MAPE-K agora

2.8.2 Arquitetura Pipe-and-Filter

Capítulo 3

Metodologia

- 3.1 Detalhamento do processo e diagramas elaborados
- 3.2 Arquitetura do código
- 3.3 Experimentos e limitações

Capítulo 4

Resultados e Discussões

Capítulo 5

Conclusões

Referências Bibliográficas

- [1] What's the difference between artificial intelligence, machine learning and deep learning? URL <https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/>.
- [2] Machine learning: o que é e qual sua importância? URL https://www.sas.com/pt_br/insights/analytics/machine-learning.html.
- [3] Aprendizado de máquina. URL https://pt.wikipedia.org/wiki/Aprendizado_de_máquina.
- [4] An architectural blueprint for autonomic computing. Technical report, IBM, 2005. URL <https://www-03.ibm.com/autonomic/pdfs/AC%20Blueprint%20White%20Paper%20V7.pdf>.
- [5] Machine learning: things are getting intense, 2018. URL <https://www2.deloitte.com/content/dam/Deloitte/global/Images/infographics/technologymediatelecommunications/gx-deloitte-tmt-2018-intense-machine-learning-report.pdf>.
- [6] Brasil se destaca com 42% das iniciativas de IA na América Latina, em 2020, 2021. URL <https://cio.com.br/tendencias/brasil-se-destaca-com-42-das-iniciativas-de-ia-na-america-latina-em-2020/>.
- [7] Proteção de Dados - LGPD, 2021. URL <https://www.gov.br/defesa/pt-br/acesso-a-informacao/lei-geral-de-protecao-de-dados-pessoais-lgpd>.
- [8] Alejandro Barredo Arrieta, Natalia Díaz Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115, 2020.
- [9] Tom Begley, Tobias Schwedes, Christopher Frye, and Ilya Feige. Explainability for fair machine learning, 2021.
- [10] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on Fairness, Accountability and Transparency*, volume 81 of *Proceedings of Machine Learning Research*, pages 77–91, 2018.

- [11] Faisal Kamiran and Toon Calders. Data pre-processing techniques for classification without discrimination. *Knowledge and Information Systems*, 2011. URL https://www.researchgate.net/publication/228975972_Data_Pre-Processing_Techniques_for_Classification_without_Discrimination.
- [12] Faisal Kamiran, Asim Karim, and Xiangliang Zhang. Decision theory for discrimination-aware classification. In *2012 IEEE 12th International Conference on Data Mining*, pages 924–929, 2012.
- [13] Andrei Kapishnikov, Tolga Bolukbasi, Fernanda B. Viégas, and Michael Terry. XRAI: better attributions through regions. In *IEEE/CVF International Conference on Computer Vision*, pages 4947–4956, 2019.
- [14] Jeffrey Kephart and D.M. Chess. The vision of autonomic computing. pages 41 – 50, 2003.
- [15] Bahador Khalegi. The how of explainable ai: Explainable modelling, 2019. URL <https://towardsdatascience.com/the-how-of-explainable-ai-explainable-modelling-55c8c43d7bed>.
- [16] Bahador Khalegi. The how of explainable ai: Post-modelling explainability, 2019. URL <https://towardsdatascience.com/the-how-of-explainable-ai-post-modelling-explainability-8b4cbc7adf5f>.
- [17] Bahador Khalegi. The how of explainable ai: Pre-modelling explainability, 2019. URL <https://towardsdatascience.com/the-how-of-explainable-ai-pre-modelling-explainability-699150495fe4>.
- [18] Sasan Maleki, Long Tran-Thanh, Greg Hines, Talal Rahwan, and Alex Rogers. Bounding the estimation error of sampling-based shapley value approximation with/without stratifying. 2013.
- [19] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328, 2017.
- [20] Brian Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. pages 335–340, 2018. URL https://www.researchgate.net/publication/330299272_Mitigating_Unwanted_Biases_with_Adversarial_Learning.