AVALIAÇÃO DE FERRAMENTAS DO ESTADO DA ARTE PARA DETECÇÃO E CORREÇÃO AUTOMÁTICA DE ERROS EM REGISTROS DE DADOS

William G. R. Medina[1], **Daniel S. Kaster**[1], **Eduardo H. M. Pena**[2]

[1]Departamento de Computação – Universidade Estadual de Londrina (UEL)
Caixa Postal 10.011 – CEP 86057-970 – Londrina – PR – Brasil

[2]Departamento de Computação – Universidade Tecnológica Federal do Paraná (UTFPR)
CEP 85902-490 – Toledo – PR – Brasil

william.medina@uel.br, dskaster@uel.br, eduardohmpena@gmail.com

Abstract. *Data collection often introduces accidental errors. Empty cells, misspellings or erroneous data, integrity and constraint violations as well as duplicate data are surprisingly common and can produce severe negative impacts on data analysis tasks. With the popularization of Data Science, the curation, unification, preparation and cleaning of data has proven to be essential in order to extract maximum value from data queries, avoiding inconsistencies and guaranteeing expected results. Numerous tools and systems have been proposed to fix these issues. Many of these have emerged in recent years, introducing a range of difficulties in operating and identifying the error detection capacity of each tool in different contexts. This work proposes the evaluation of each method taking into account different scenarios, bringing together a combination of methods from recent literature.*

Resumo. *A coleta de dados frequentemente introduz erros acidentais. Células vazias, grafias ou digitações errôneas, violações de regras de integridade e de negócios e duplicações involuntárias de informações são surpreendentemente comuns e podem ter severos impactos negativos em tarefas de análises de dados. Com a popularização da área de Data Science, a curadoria, unificação, aprestamento e limpeza de dados têm se mostrado essenciais para extrair o máximo valor em levantamentos de qualquer natureza, evitando inconsistências e garantindo a manipulação esperada. Vários métodos e sistemas de detecção de erros se propuseram a resolver os desafios intrínsecos à correção de dados. Muitos destes métodos surgiram nos últimos anos, introduzindo dificuldades na operação e identificação da capacidade de detecção de erros de cada um em diferentes contextos. Este trabalho propõe a avaliação destes métodos e ferramentas levando-se em conta diferentes cenários, trazendo uma combinação dos métodos da literatura do estado da arte.*

1. Introdução

Empresas e instituições em geral vêm coletando grandes quantidades de dados de uma variedade de fontes complementares. Os repositórios de dados gerados têm como ob-

jetivo final alimentar tarefas analíticas, possibilitando integrar dados sob diferentes aspectos para gerar informações com valor agregado para o propósito da organização. A manipulação de diferentes fontes de dados frequentemente introduz erros que podem causar problemas em projetos analíticos, particularmente naqueles que se utilizam de técnicas de aprendizado de máquina. Krishman, S.; Wang, J.; Wu, E. et. al [12] apontam que a corrupção sistemática em uma só variável é o suficiente para causar modelos deslocados, alguns a ponto de trazer resultados decrescentes de aprendizado em modelos que se utilizam de machine learning. Por conta destes problemas, a curadoria de dados têm se mostrado cada vez mais em pauta para tarefas analíticas.

Uma pesquisa com cientistas de dados feita pela companhia americana de Inteligência Artificial (IA) CrowdFlower (atualmente denominada Figure Eight) e publicada no website da revista Forbes [1] exibiu que mais de 60% do tempo investido em projetos de ciência de dados é destinada à limpeza e organização de dados. Além deste fator, 57% dos cientistas de dados alegaram que a limpeza e organização dos dados é a parte menos agradável do seu trabalho.

Limpeza de dados tipicamente compreende duas fases: (1) detecção de erros, onde erros de tipos variados são identificados, e (2) reparo de erros, em que atualizações são aplicadas aos dados, automaticamente ou por sugestão de usuários especialistas, gerando uma versão mais "limpa" do conjunto de dados [10]. Técnicas de detecção de erros podem ser quantitativas ou qualitativas. Técnicas quantitativas frequentemente utilizam métodos estatísticos para identificar valores anormais ou extremos, que diferem muito do comportamento do restante do conjunto de dados. Por exemplo, um resultado de exame laboratorial com valor muito acima dos valores máximos tipicamente identificados deve ser um erro. Já técnicas qualitativas de detecção de erros baseiam-se em abordagens que identificam padrões ou restrições de integridade válidas para uma instância consistente de um conjunto de dados e reportam como erros os valores que violam tais padrões ou restrições [10].

Nos últimos anos, várias ferramentas comerciais de limpeza de dados foram desenvolvidas, no entanto, a grande maioria destas ferramentas se limitam a métodos que exigem algum grau de conhecimento do usuário a despeito de regras de dependência e de negócio referentes ao conjunto em análise, ou apresentam grau de detecção de erros insatisfatório [2]. Outro problema é que dependendo do conjunto em análise, resultados de um mesmo algoritmo ou ferramenta podem apresentar taxas de precisão e/ou revocação discrepantes a níveis acima de 90% em alguns casos, como mostram Adebjan Z., Chu X., Deng D., et al. [2] quando analisam a ferramenta TRIFACTA [2] em seus conjuntos de dados "MIT VPF" e "Animal".

Um estudo de 2016 [2] compara o desempenho de ferramentas e algoritmos de limpeza de dados presentes naquela época. Porém, desde então houve avanço significativo na área de limpeza de dados, particularmente na literatura. Propostas com a utilização de algoritmos de aprendizado de máquina têm sido apresentadas e em boa parte dos casos têm superado níveis de precisão e revocação de propostas usuais. Por exemplo, HoloDetect [9] apresenta a ideia de um algoritmo que, utilizando-se de técnicas de *few-shot*

[1] https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/

[2] https://www.trifacta.com/

learning, apresenta taxas de precisão e revocação de 0.903 e 0.989, respectivamente, em comparação a taxas de 0.64 e 0.667, respectivamente, quando comparadas a técnicas de detecção de outliers - amplamente empregadas em ferramentas como dboost [19].

Nas propostas de ferramentas e algoritmos de detecção de erros em bases de dados na literatura, cada uma utiliza seus próprios conjuntos de dados para sustentação de efetividade. Mediante este fato, o usuário que deseja utilizar alguma destas ferramentas se vê impossibilitado de decidir qual ferramenta utilizar. O propósito deste trabalho é a análise experimental e atualizada do estado da arte em detecção automática de erros, sistematizando e mensurando o potencial de cada ferramenta para o campo de limpeza de dados com conjuntos variados de dados. Com a reunião de um conjunto de dados teste, este estudo medirá a aptidão de cada ferramenta no que tange às suas técnicas de detecção utilizadas, taxas de precisão e revocação, dentre outras métricas de desempenho.

Este trabalho está dividido da seguinte forma: a seção 2 apresenta fundamentos importantes da área de limpeza de dados e algumas das técnicas empregadas atualmente nesta area; a seção 3 descreve os objetivos deste trabalho; a seção 4 apresenta os procedimentos metodológicos que serão aplicados; a seção 5 mostra o cronograma de execução; a seção 6 apresenta as contribuições e resultados esperados deste trabalho.

2. Fundamentação Teórico-Metodológica e Estado da Arte

As subseções seguintes irão apresentar os conceitos necessários para a compreensão da proposta do trabalho, de forma a caracterizar o problema sobre o qual é fundamentado.

2.1. Limpeza de dados

A limpeza de dados consiste na implementação de estratégias para evitar ou corrigir erros, garantindo integridade da informação em bancos de dados relacionais. A figura 1 indica o típico fluxo de uma tarefa de limpeza de dados.

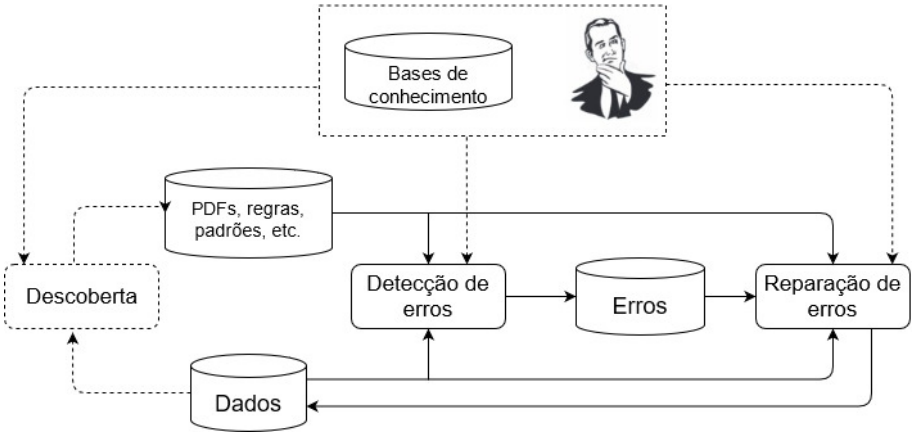

Figura 1. Tarefa de limpeza de dados. Figura retirada de [10]

Concentrando-se na etapa de detecção de erros, dado um conjunto de dados, varreduras são executadas de forma a identificar erros. Uma proposta simples é a consulta à um especialista em domínio, que partindo de bases de conhecimento consegue identificar problemas. Por se tratar de um método manual de detecção exige tempo, consome recursos e tem relação unívoca com o tamanho do conjunto analisado. Para contornar este

problema, propostas com a utilização de técnicas de data profiling têm vindo à tona e se mostrado promissoras [13], pois possibilitam a detecção automática de metadados.

Metadados referem-se a "dados sobre dados"[1]. Seja a coluna CEP de um conjunto de endereços X sobre o Brasil, a coluna que guarde os CEPs de cada endereço. Em uma análise cuidadosa, nota-se que todos os CEPs do estado do Paraná estão na faixa de 80000-000 a 87999-999. Nesse sentido, a afirmação "todos os CEPs do Paraná estão na faixa de 80000-000 a 87999-999"pode ser considerada como metadado sobre os CEPs do Brasil. A etapa responsável pela descoberta de regras e metadados sobre o conjunto analisado é representada pelo passo de descoberta.

Uma vez que se conheça informações sobre metadados de um conjunto e/ou se detectem erros, é possível prosseguir à etapa de confirmação e reparação de erros. No caso de detecções automáticas de erros, há a acusação dos mesmos quando uma informação não condiz com seu respectivo metadado, ou estima-se que a informação esteja incorreta. Através da utilização de técnicas de limpeza de dados, tal como a detecção de valores outliers, isto é, valores muito discrepantes quando comparados a seus similares, é possível a identificação de valores que não correspondem ao esperado.

Um problema inerente à detecção automática de erros é a detecção eficiente. Em uma análise perfeita, espera-se que as taxas de precisão e de revocação estejam ambas em 1, indicando que a cada cem erros detectados, cem destes são realmente erros, e a cada cem erros presentes no conjunto de dados, cem destes foram detectados. Porém, estudos mostram que, mesmo com a combinação de várias ferramentas, taxas de precisão e revocação tão baixas quanto 0.33 e 0.575 foram identificadas em algumas bases de dados [2], indicando um longo caminho a ser percorrido para taxas de desempenho satisfatório.

2.2. Métricas de avaliação

Existem três métricas muito utilizadas para avaliar a eficiência de algoritmos de limpeza de dados.

Precisão é uma métrica que avalia a proporção de identificações que estão realmente corretas dentre um conjunto apontado como correto. Formalmente, ela pode ser definida pela função

$$\frac{VP}{VP + FP} \tag{1}$$

onde VP é a quantidade de verdadeiros positivos do conjunto e FP é a quantidade de falsos positivos do conjunto. Uma precisão de 0.33 indica que a cada 100 células apontadas como erro, apenas 33 são realmente erros.

Revocação é uma métrica que avalia a proporção de identificações corretas dentre todas as corretas possíveis. Matematicamente, a taxa de revocação pode ser definida pela fórmula

$$\frac{VP}{VP + FN} \tag{2}$$

Onde VP é a quantidade de verdadeiros positivos do conjunto e FN é a quantidade de falsos negativos do conjunto. Uma taxa de revocação de 0.58 indica que dentre 100 erros em células, apenas 58 destes são detectados.

Por fim, a taxa de medida-F é uma métrica que avalia a acurácia de um teste. Ela é calculada a partir da precisão e taxa de revocação de um teste, sendo a média harmônica

de ambas. Matematicamente, ela é definida pela função

$$2 \cdot \frac{P \cdot R}{P + R} \tag{3}$$

onde P é a taxa de precisão e R é a taxa de revocação.

2.3. Data profiling

Data profiling refere-se à atividade de colecionar dados sobre dados, isto é, metadados [3]. A preparação e descobrimento de dados é essencial ao processo de limpeza de dados, pois proporciona uma série de métodos para detecção e correção de erros. Enquanto as ofertas comerciais existentes no espaço de detecção de erros estão em sua maioria limitados a regras manualmente definidas, na literatura há uma variedade de técnicas propostas para a detecção de metadados automática, particularmente nos últimos anos [6][4][17][18]. Técnicas de data profiling se dividem em três categorias principais; duas serão aqui abordadas: metadados de coluna única e metadados de dependência.

2.4. data profiling com dados de coluna única

Erros de coluna única referem-se àqueles que não possuem dependências funcionais com outras colunas. Valores duplicados em uma coluna com restrição de unicidade ou valores vazios em colunas de preenchimento obrigatório são exemplos de erros de coluna única, pois independem de outras colunas para serem identificadas como erros. Estes equívocos apresentam um grande desafio para analistas de dados, principalmente àqueles que utilizam de modelos baseados em aprendizado de máquina, pois podem gerar modelos destorcidos [12]. A figura 2 indica os efeitos de análises de aprendizado de máquina em modelos com dados sujos, limpos e sujos e totalmente limpos com poucas amostras.

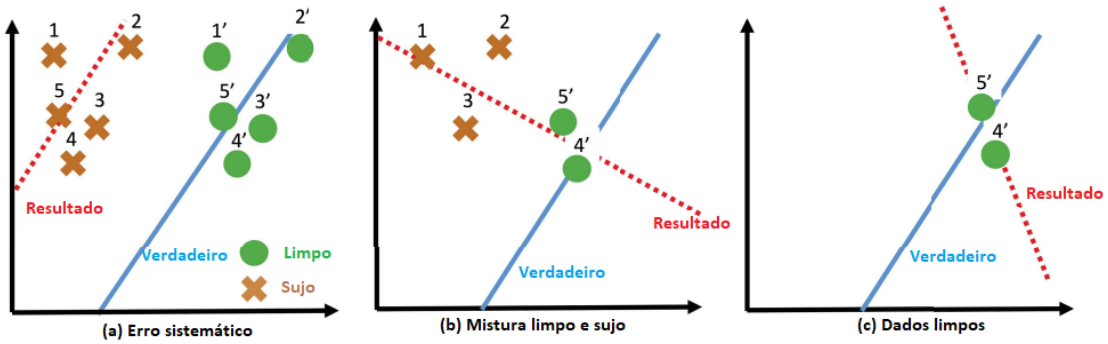

Figura 2. (a) Corrupção sistemática de uma variável pode levar a um modelo deslocado. Os exemplos sujos estão etiquetados de 1-5 e os exemplos limpos de 1' a 5'. (b) A mistura de dados limpos e sujos resulta em um modelo de menos acurácia que nenhuma limpeza. (c) Pequenas amostras de apenas dados limpos pode resultar em problemas similares. Figura retirada de [12].

Para tratar incorreções de coluna única como erros de digitação, uma diversidade de algoritmos e soluções foram propostas. Exemplos incluem a ferramenta OpenRefine [11] que se utiliza de algoritmos de cluster para verificação de similaridade e agrupa os dados utilizando métodos de vizinho mais próximo para sugerir ao usuário possíveis representações de dados que se tratam da mesma entidade, e Dboost [19], um algoritmo veloz que detecta valores outliers em um conjunto de dados, facilitando a detecção de possíveis erros.

2.5. data profiling com dependências

Algoritmos de detecção de erros em coluna única conseguem tratar problemas como erros de digitação e valores faltantes, mas são desprovidos de detecções de erros estruturais. Como resultado, boa parte de erros não são detectados.

Dependências são metadados que descrevem relações entre colunas [3]. Exemplos típicos incluem dependências funcionais e dependências de inclusão. Uma dependência funcional é uma relação entre dois atributos dentro de uma mesma tabela. Por exemplo, dado o CEP de um endereço, espera-se que este se refira a determinada cidade e rua específica. Portanto, a coluna CEP possui dependência funcional com as colunas rua e cidade. Dependências de inclusão são instruções na qual algumas colunas de uma relação estão contidas em outras colunas, tipicamente através de chave estrangeira. Por exemplo, dado uma tabela "peças"e outra tabela "peças-fornecedor", ambas com os atributos "número-da-peça"referindo-se à mesma entidade, diz-se que essas duas colunas possuem dependências de inclusão entre si, pois a inserção em uma tabela implica na existência e/ou inserção na outra.

Idealmente, restrições estruturais em bases de dados são conhecidas no momento de sua criação, possibilitando ao programador especificar restrições evitando inserções que violem regras de negócio. No entanto, muitas bases de dados não vêm com restrições de dependência explícitas ou conhecidas no momento de sua criação, viabilizando erros. Ferramentas como DC-Clean [7] oferecem ao usuário a possibilidade de inserção de regras de qualidade e restrições de negação, porém necessitam que o usuário as conheça previamente. Outras, como KATARA [8], buscam verificações semânticas utilizando-se de bases externas de conhecimento e contam com validação humana para o reconhecimento de dependências. Também recentemente, mostrou-se que restrições de negação possuem maior capacidade de representação do que restrições de dependências funcionais, condicionais ou não. Com esta descoberta, surgiram ferramentas mais capazes de automatização da descoberta de dependências [6].

2.6. Aprendizado de máquina

O aprendizado de máquina diz respeito ao método computacional que, dado um conjunto de dados de teste e treinamento, possibilita o aprimoramento de uma dada tarefa computacional [5]. Essa tarefa geralmente consiste em realizar alguma previsão. Por exemplo, suponha que em um programa computacional foram analisadas as características faciais de jovens entre 18 a 25 anos. Dada a tarefa de determinar se um rosto combina com um jovem, um algoritmo de aprendizado de máquina pode – usando as informações coletadas – prever se a pessoa é jovem, mesmo que nunca tenha sido vista, pois suas características esperadas são conhecidas.

2.7. Aprendizado de máquina e limpeza de dados

Estudos têm mostrado que não existe uma única melhor aplicação de métodos de limpeza devido às dependências relacionais de dados. Mesmo quando várias ferramentas são aplicadas, o desempenho é frequentemente ruim [2] porque se não foi capturada a natureza holística do processo de limpeza. Por isso, técnicas que se utilizam de aprendizado de máquina têm apresentado uma direção interessante de linha de pesquisas, particularmente para a preparação e limpeza de dados. ActiveClean [12] propõe uma abordagem baseada

em aprendizado de máquina para a resolução de erros em conjuntos de dados. Através de métodos estatísticos, particularmente valores *term frequency–inverse document frequency (TF-IDF)*, a ferramenta treina classificadores que podem identificar tuplas contendo erros. Mais recentemente, propostas como HoloDetect [9] propuseram técnicas de *few-shot learning* para detecção e correção de erros, com valores de precisão e revocação em torno de 90% em alguns conjuntos sintéticos de dados. Tais propostas necessitam de ajustes efetuados pelo usuário, como a seleção ou configuração do algoritmo. Para limitar ao máximo a necessidade de interação humana, Mohammad M., Ziawasch A., Raul C. F., et al. propõem um algoritmo livre de configurações, de forma semi-supervisionada [15] com a utilização das ferramentas Raha [16] e Baran [14]. Primeiro, cada detector / corretor de erro de base gera um conjunto inicial de erros / correções de dados em potencial. Esta etapa aumenta particularmente o limite de recuperação alcançável da tarefa de detecção / correção de erros. Em seguida, Raha/Baran agrupa a saída desses detectores / corretores de erros básicos em um conjunto final de erros / correções de dados de uma maneira semissupervisionada. Raha / Baran pede iterativamente ao usuário para anotar uma tupla, ou seja, marcar / corrigir alguns erros de dados. Raha / Baran então aprende a generalizar os exemplos de detecção / correção de erros fornecidos pelo usuário para o resto do conjunto de dados, de acordo, sem auxílio do usuário.

Mesmo com o avanço de pesquisas na área de aprendizado de máquina, há ainda problemas que se destacam. Por instância, sabe-se que algoritmos de aprendizado de máquina necessitam de certa relevância entre dados. No entanto, a natureza flexível das grandes empresas e negócios implica em mudanças constantes em dados. Dados que possuem muita maleabilidade constituem barreiras para a criação de algoritmos com base em aprendizado de máquina pois a detecção de itens corretos e incorretos é uma tarefa não trivial. Além do mais, existem conjuntos de dados que exigem conhecimento estrutural da organização dos dados, de forma que usuários leigos não conseguiriam identificar problemas, aumentando a dificuldade de criação de algoritmos de aprendizado de máquina.

3. Objetivos

Realizar um estudo sobre sistemas de detecção e correção de erros em bancos de dados, particularmente propostas mais recentes e não muito difundidas na área.

4. Procedimentos metodológicos/Métodos e técnicas

Será feito um levantamento bibliográfico sobre algoritmos de detecção e correção de dados do estado da arte. Em seguida, será feita a obtenção de ferramentas ou repositórios de códigos de detecção de erros, bem como a preparação e levantamento de dados de teste, propiciando a análise de cada ferramenta quanto às suas capacidades e deficiências em detecção e correção de erros.

Serão medidas as taxas de precisão, revocação e medida-F de cada ferramenta que se encontre em disponibilidade para cada base de dados teste levantada. Estas ferramentas serão comparadas de forma a se caracterizar os defeitos e vantagens de cada uma e, no caso de discrepâncias significativas, far-se-ão análises buscando o motivo pelo o qual o desempenho foi ou não insuficiente comparado a outras ferramentas ou bases de dados.

No caso de ferramentas que se utilizam de múltiplos recursos de limpeza de dados, será feito um levantamento de cada algoritmo que a ferramenta utiliza, caracterizando os

diferentes tipos de erros aos quais a ferramenta está ou não apta a resolver de forma eficaz.

5. Cronograma de Execução

Atividades:

1. Estudo dos fundamentos de limpeza de dados;
2. Estudo das principais técnicas da literatura de detecção automática de erros, em particular, de propostas mais recentes e ainda menos conhecidas na área;
3. Levantamento e estudo de ferramentas com métodos de detecção de erros, de acesso livre ou com versões de avaliação, e implementações específicas de métodos;
4. Levantamento de conjuntos de dados variados, da literatura de detecção de erros ou preparados durante o projeto, para realizar a análise experimental;
5. Preparação de casos de teste para execução nas diferentes ferramentas escolhidas;
6. Execução dos casos de teste e coleta de resultados;
7. Escrita do Trabalho de Conclusão de Curso.

Tabela 1. Cronograma de Execução

	ago	set	out	nov	dez	jan	fev	mar	abr	mai	jun	jul
Atividade 1	X	X	X									
Atividade 2			X	X	X							
Atividade 3					X	X	X	X				
Atividade 4							X	X				
Atividade 5								X	X	X	X	
Atividade 6									X	X	X	X
Atividade 7							X	X	X	X	X	X

6. Contribuições e/ou Resultados esperados

Dentre os resultados esperados deste trabalho, destacam-se os seguintes.

- Demonstrar a capacidade de cada ferramenta de detecção de erros quando comparadas levando-se em conta diferentes contextos;
- apresentar os desafios inerentes a cada ferramenta, fornecendo ao leitor uma visão atualizada dos algoritmos de erros em bancos de dados;
- proporcionar a fácil identificação de qual ferramenta utilizar a pesquisadores que se encontram em dúvidas de qual ferramenta é melhor para a sua tarefa de limpeza de dados;
- demonstrar se alternativas de aprendizado de máquina constituem via universal para tarefas de limpeza de dados, ou se alternativas usuais ainda apresentam contribuições importantes e indispensáveis.

7. Espaço para assinaturas

Londrina, *treze de setembro de 2021*.

_____ _____

Aluno Orientador

Referências

[1] Merriam webster. `https://www.merriam-webster.com/dictionary/metadata`. Accessed: 2021-09-13.

[2] Ziawasch Abedjan, Xu Chu, Dong Deng, Raul Castro Fernandez, Ihab F. Ilyas, Mourad Ouzzani, Paolo Papotti, Michael Stonebraker, and Nan Tang. Detecting data errors: Where are we and what needs to be done? *Proc. VLDB Endow.*, 9(12):993–1004, August 2016.

[3] Ziawasch Abedjan, Lukasz Golab, Felix Naumann, and Thorsten Papenbrock. 2018.

[4] Tobias Bleifuß, Sebastian Kruse, and Felix Naumann. Efficient denial constraint discovery with hydra. *Proc. VLDB Endow.*, 11(3):311–323, November 2017.

[5] A. Burkov. *The Hundred-Page Machine Learning Book*. Andriy Burkov, 2019.

[6] Xu Chu, Ihab F. Ilyas, and Paolo Papotti. Discovering denial constraints. *Proc. VLDB Endow.*, 6(13):1498–1509, August 2013.

[7] Xu Chu, Ihab F. Ilyas, and Paolo Papotti. Holistic data cleaning: Putting violations into context. In *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, pages 458–469, 2013.

[8] Xu Chu, John Morcos, Ihab F. Ilyas, Mourad Ouzzani, Paolo Papotti, Nan Tang, and Yin Ye. Katara: A data cleaning system powered by knowledge bases and crowdsourcing. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, SIGMOD '15, page 1247–1261, New York, NY, USA, 2015. Association for Computing Machinery.

[9] Alireza Heidari, Joshua McGrath, Ihab F. Ilyas, and Theodoros Rekatsinas. Holodetect: Few-shot learning for error detection. In *Proceedings of the 2019 International Conference on Management of Data*, SIGMOD '19, page 829–846, New York, NY, USA, 2019. Association for Computing Machinery.

[10] Ihab F. Ilyas and Xu Chu. *Data Cleaning*. Association for Computing Machinery, New York, NY, USA, 2019.

[11] Ham K. Openrefine (version 2.5). http://openrefine.org. free, open-source tool for cleaning and transforming data. *J Med Libr Assoc.*, 2013.

[12] Sanjay Krishnan, Jiannan Wang, Eugene Wu, Michael J. Franklin, and Ken Goldberg. Activeclean: Interactive data cleaning for statistical modeling. *Proc. VLDB Endow.*, 9(12):948–959, August 2016.

[13] Tien Fabrianti Kusumasari and Fitria. Data profiling for data quality improvement with openrefine. In *2016 International Conference on Information Technology Systems and Innovation (ICITSI)*, pages 1–6, 2016.

[14] Mohammad Mahdavi and Ziawasch Abedjan. Baran: Effective error correction via a unified context representation and transfer learning. *Proc. VLDB Endow.*, 13(12):1948–1961, July 2020.

[15] Mohammad Mahdavi and Ziawasch Abedjan. Semi-supervised data cleaning with raha and baran. In *CIDR*, 2021.

[16] Mohammad Mahdavi, Ziawasch Abedjan, Raul Castro Fernandez, Samuel Madden, Mourad Ouzzani, Michael Stonebraker, and Nan Tang. Raha: A configuration-free error detection system. In *Proceedings of the 2019 International Conference on Management of Data*, SIGMOD '19, page 865–882, New York, NY, USA, 2019. Association for Computing Machinery.

[17] Eduardo H. M. Pena, Eduardo C. de Almeida, and Felix Naumann. Discovery of approximate (and exact) denial constraints. *Proc. VLDB Endow.*, 13(3):266–278, November 2019.

[18] Eduardo H. M. Pena and Eduardo Cunha de Almeida. Bfastdc: A bitwise algorithm for mining denial constraints. In Sven Hartmann, Hui Ma, Abdelkader Hameurlain, Günther Pernul, and Roland R. Wagner, editors, *Database and Expert Systems Applications*, pages 53–68, Cham, 2018. Springer International Publishing.

[19] Yuxiao Zhang, Xiaorong Wang, Bingyang Li, Wei Chen, Tengjiao Wang, and Kai Lei. Dboost: A fast algorithm for dbscan-based clustering on high dimensional data. In James Bailey, Latifur Khan, Takashi Washio, Gill Dobbie, Joshua Zhexue Huang, and Ruili Wang, editors, *Advances in Knowledge Discovery and Data Mining*, pages 245–256, Cham, 2016. Springer International Publishing.