UNIVERSIDADE FEDERAL DO MARANHÃO

Programa de Pós-Graduação em Engenharia Elétrica

Antonio Fernando Lavareda Jacob Junior

Algoritmos Genético para Imputação Múltipla de Dados na Classificação Multirrótulo

São Luís - MA

2024

Antonio Fernando Lavareda Jacob Junior

Algoritmos Genético para Imputação Múltipla de Dados na Classificação Multirrótulo

Tese apresentada como requisito parcial para obtenção do título de Doutor em Engenharia Elétrica, ao Programa de Pós-Graduação em Engenharia Elétrica, da Universidade Federal do Maranhão.

Programa de Pós-Graduação em Engenharia Elétrica

Universidade Federal do Maranhão - UFMA

Orientador: Prof. Dr. Ewaldo Eder Carvalho Santana

Coorientador: Prof. Dr. Fábio Manoel França Lobato

São Luís - MA

2024

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).
Diretoria Integrada de Bibliotecas/UFMA

Jacob Junior, Antonio Fernando Lavareda.
Algoritmos Genético para Imputação Múltipla de Dados na Classificação Multirrótulo / Antonio Fernando Lavareda Jacob Junior. - 2024.
97 p.

Orientador(a): Ewaldo Eder Carvalho Santana Fábio Manoel França Lobato.
Tese (Doutorado) - Programa de Pós-graduação em Engenharia Elétrica/ccet, Universidade Federal do Maranhão, São Luís - MA, 2024.

1. Algoritmos genéticos. 2. Classificação multirrótulo. 3. Valores Ausentes. I. Fábio Manoel França Lobato, Ewaldo Eder Carvalho Santana. II. Título.

Antonio Fernando Lavareda Jacob Junior

Algoritmos Genético para Imputação Múltipla de Dados na Classificação Multirrótulo

Tese apresentada como requisito parcial para obtenção do título de Doutor em Engenharia Elétrica, ao Programa de Pós-Graduação em Engenharia Elétrica, da Universidade Federal do Maranhão.

São Luís - MA, 23 de fevereiro de 2024:

Prof. Dr. Ewaldo Eder Carvalho Santana
Orientador - UFMA

Prof. Dr. Fábio Manoel França Lobato
Coorientador - Universidade Federal do Oeste do Pará - UFOPA

Prof. Dr. Allan Kardec Duailibe Barros Filho
Examinador Interno - UFMA

Prof. Dr. Francisco Jose Da Silva E Silva
Examinador Interno - UFMA

Prof. Dr. Omar Andres Carmona Cortes
Examinador Externo - IFMA

Prof. Dr. Marcelino Silva da Silva
Examinador Externo - UFOPA

Dedico esse trabalho aos meus familiares
e amigos que sempre me incentivaram

Agradecimentos

Em primeiro lugar, expresso minha profunda gratidão a Deus, fonte de força e inspiração ao longo desta jornada acadêmica.

Aos meus orientadores, Professores Ewaldo Santana e Fábio Lobato, reconheço a importância vital de suas orientações sábias e dedicadas. Sua expertise e incentivo foram determinantes para o desenvolvimento deste trabalho, e por isso, sou imensamente grato.

Aos meus pais e familiares, meu eterno agradecimento. O apoio incondicional e o amor que sempre demonstraram foram pilares essenciais durante os desafios deste percurso.

À minha esposa Carolina e ao meu filho Henrique, agradeço pela compreensão, paciência e amor incondicional. Seu apoio constante foi a luz que guiou meus passos nos momentos mais desafiadores.

Não poderia encerrar meus agradecimentos sem expressar minha profunda gratidão a todos os meus alunos e orientandos. Sua dedicação, entusiasmo e contribuições foram elementos-chave para o êxito desta jornada acadêmica. Cada um de vocês, de maneira única, deixou uma marca valiosa no desenvolvimento deste trabalho e da minha jornada.

Em especial, quero estender meus agradecimentos a Fabrício Almeida. Sua colaboração incansável, habilidades excepcionais e comprometimento foram fundamentais para o progresso deste projeto. Agradeço por sua parceria e pelo valioso tempo e esforço dedicados a esta pesquisa.

Gostaria, também, de agradecer o financiamento parcial deste trabalho pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) por meio processo 147336/2020-1.

Por último, dedico este trabalho ao meu querido tio Francisco Cardoso. Sua memória permanece viva em meu coração, e sua torcida constante por este momento é uma lembrança que carregarei para sempre. Seu espírito incentivador e carinho serão eternamente lembrados e celebrados neste importante capítulo da minha vida acadêmica.

"Nosso céu tem mais estrelas,
Nossas várzeas têm mais flores,
Nossos bosques têm mais vida,
Nossa vida mais amores."

(Gonçalves Dias)

Resumo

Dados ausentes são um problema prevalente que requer atenção, uma vez que a maioria das técnicas de análise de dados não consegue lidar com isso. Esse problema é particularmente crítico em Classificação Multi-rótulo (MLC), onde poucos estudos têm investigado dados ausentes nesse domínio de aplicação. MLC difere da Classificação de Monorrótulo (SLC) ao permitir que uma instância seja associada a várias classes. A classificação de filmes é um exemplo didático, já que um filme pode ser classificado como "drama" e "biografia" simultaneamente. Um dos métodos mais comuns de tratamento de dados ausentes é por meio da imputação de dados, a qual busca valores plausíveis para preencher os ausentes. Nesse cenário, essa tese apresenta um novo método de imputação baseado em um algoritmo genético multiobjetivo para otimizar múltiplas imputações de dados, chamado Imputação Múltipla de Dados na Classificação Multirrótulo por meio de um Algoritmo Genético, ou simplesmente EvoImp. Aplicamos o método proposto em aprendizado multirrótulo e avaliamos seu desempenho usando seis bancos de dados sintéticos, considerando vários cenários de distribuição de valores ausentes. O método foi comparado com outras estratégias de imputação do estado-da-arte, como *K-Means Imputation* (KMI) e *Weighted K-Nearest Neighbors Imputation* (WKNNI). Os resultados comprovaram que o método proposto superou o *baseline* em todos os cenários, alcançando as melhores medidas de avaliação considerando: *Exact Match*, Acurácia e *Hamming Loss*. Os resultados superiores foram consistentes em diferentes domínios e tamanhos de conjuntos de dados, demonstrando a robustez do EvoImp. Assim, o EvoImp representa uma solução viável para o tratamento de dados ausentes em aprendizado multirrótulo.

Palavras-chave: Valores Ausentes, Classificação Multirrótulo, Algoritmos Genéticos.

Abstract

Missing data is a prevalent problem that requires attention, as most data analysis techniques are unable to handle it. This is particularly critical in Multi-Label Classification (MLC), where only a few studies have investigated missing data in this application domain. MLC differs from Single-Label Classification (SLC) by allowing an instance to be associated with multiple classes. Movie classification is a didactic example since it can be "drama" and "bibliography" simultaneously. One of the most usual missing data treatment methods is data imputation, which seeks plausible values to fill in the missing ones. In this scenario, we propose a novel imputation method based on a multi-objective genetic algorithm for optimizing multiple data imputations called Multiple Imputation of Multi-label Classification data with a genetic algorithm, or simply EvoImp. We applied the proposed method in multi-label learning and evaluated its performance using six synthetic databases, considering various missing values distribution scenarios. The method was compared with other state-of-the-art imputation strategies, such as K-Means Imputation (KMI) and weighted K-Nearest Neighbors Imputation (WKNNI). The results proved that the proposed method outperformed the baseline in all the scenarios by achieving the best evaluation measures considering the Exact Match, Accuracy, and Hamming Loss. The superior results were constant in different dataset domains and sizes, demonstrating the EvoImp robustness. Thus, EvoImp represents a feasible solution to missing data treatment for multi-label learning.

Keywords: Missing values, Multi-Label Classification, Genetic Algorithms.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

ACC	*Accuracy*
AG	Algoritmo Genético
AM	Aprendizagem de Máquina
BR	*Binary Relevance*
CC	*Classifier Chains*
CMC	*Concept Most Common*
CRISP-DM	*Cross Industry Standard Process for Data Mining*
ECC	*Ensembles of Classifier Chains*
EM	*Exact Match*
IA	Inteligência Artificial
IoT	*Internet of Things*
IM	Imputação Múltipla
IS	Imputação Simples
HL	*Hamming loss*
HOMER	*Hierarchy of Multi-label classifiER*
KDD	*Knowledge Discovery in Databases*
KMI	*K-Means clustering Imputation*
KNN	*K-Nearest Neighbors*
KNNI	*K-Nearest Neighbors Imputation*
MAP	*Maximum A Posteriori*
MAR	*Missing At Random*
MC	*Most Common*
MD	Mineração de Dados

MCAR	*Missing Completely At Random*
micro-GA	*micro-genetic algorithms*
ML	*Missing Label*
MNAR	*Missing Not At Random*
ML-kNN	*Multi-label K-Nearest Neighbors*
MLC	*Multi-label Classification*
NRMSE	*Normalized Root-Mean-Square Error*
ROC	*Receiver Operating Characteristic*
RMSE	*Root-Mean-Square Error*
SLC	*Single-label Classification*
VAs	Valores Ausentes
WKNNI	*Weighted Imputation with K-Nearest Neighbor*

Sumário

1 INTRODUÇÃO

Com o avanço da tecnologia e o crescente volume e variedade de dados disponíveis, torna-se cada vez mais difícil o processamento e obtenção de *insights* significati-vos de maneira manual, e em alguns casos, até ultrapassa a capacidade dos bancos de dados convencionais (PROVOST; FAWCETT, 2013; SESTINO et al., 2020). A Mineração de Dados (MD) tem sido reconhecida como uma tarefa importante e desafiadora ao lidar com diversos problemas do dia a dia (TSAI; LI; LIN, 2018), tais como: identificação de tendências para prever epidemias (SINGH; SINGH, 2023); segmentação de mercado (REY-BLANCO et al., 2024); análise de sentimentos em redes sociais para entender o comportamento do consumidor (CIRQUEIRA et al., 2020); otimização de rotas de transporte (BAI et al., 2023); previsão de mudanças climáticas (SA'ADI et al., 2023); dentre outros. Nesse sentido, alguns estudos concentram-se nos desafios enfrentados por empresas ao implementar técnicas destinadas a superar essas questões (TCHUENTE; HADDADI, 2023; ABOU-FOUL; RUIZ-ALBA; LÓPEZ-TENORIO, 2023).

Nesse contexto, tem-se por objetivo da MD a extração de padrões úteis e conhecimento de grandes bancos de dados (ARRIETA et al., 2020; SHU; YE, 2023). No entanto, os pesquisadores ainda não chegaram a um consenso sobre as limitações conceituais do termo "Mineração de Dados"; alguns afirmam que a MD é apenas uma etapa no processo de descoberta de conhecimento em bancos de dados - *Knowledge Discovery in Databases* (KDD), enquanto outros vinculam o termo a todo o processo de extração de conhecimento (HAN; KAMBER; PEI, 2012; FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996).

No entanto, todos corroboram que o processo deve ser dividido em diferentes estágios devido à sua complexidade. Rezende (2003) lista quatro fases básicas: identificação do problema, pré-processamento, extração de padrões e pós-processamento. A fase de pré-processamento, responsável por preparar e selecionar dados, é mais propensa a erros, e vários autores afirmam que é responsável por até 80% de todo o processo de descoberta de informações (PRESS, 2016). Nesse caso, a seleção do conjunto de dados é um dos primeiros passos no processo e pode ajudar a reduzir o esforço necessário na fase de pré-processamento (LIN; TSAI, 2020).

Todavia, na prática, *datasets* do mundo real contém uma proporção de instâncias com Valores Ausentes (VAs), os quais são ubíquos na análise de dados (HEYMANS; TWISK, 2022). Como forma de exemplificar essa pluralidade de contexto que podem ser encontrados valores ausentes, os seguintes trabalhos e áreas podem ser destacados: saúde e medicina (APPELBAUM et al., 2023; ZHANG et al., 2023); engenharia de transporte (ZUO et al., 2023; KONG et al., 2023); *Internet of Things* (IoT) (LI et al., 2023); monitoramento

ambiental (MORE; WOLKERSDORFER, 2023); desempenho financeiro e ambiental, social e de governança (*Environmental, Social and Governance* - ESG) (EKINCI et al., 2024); segurança da informação (SZCZEPAŃSKI et al., 2023); etc.

As causas dos VAs são as mais diversas e relacionadas ao domínio de aplicação (SANTOS et al., 2019). Isso inclui desvantagens na aquisição de dados, erros de medição, problemas em redes de sensores, falhas na migração de dados e relutância em responder a perguntas de pesquisa (HONAKER; KING, 2010; TSAI; LI; LIN, 2018).

Como algoritmos/métodos de análise de dados não são projetados para lidar com valores ausentes, é essencial tratá-los antes, a fim de garantir a validade dos resultados, para não prejudicar as conclusões da pesquisa (HEYMANS; TWISK, 2022; LIN; TSAI, 2020; GARCIARENA; SANTANA, 2017). VAs são problemáticos devido ao risco de viés, o qual depende do tipo de dados ausentes, da extensão da ausência e de como lidar com os VAs nas análises (HEYMANS; TWISK, 2022). Portanto, é crucial lidar com os dados ausentes de maneira oportuna para tomadas de decisão inteligentes (ADHIKARI et al., 2022).

Diversas técnicas surgiram para lidar com esse problema (LUENGO; GARCÍA; HERRERA, 2012; LIN; TSAI, 2020; EMMANUEL et al., 2021). Lin e Tsai (2020) comentam que se a taxa de VAs for inferior a 10% ou 15%, eles podem ser removidos sem causar perda significativa ao processo de mineração de dados. No entanto, isso não significa que os conjuntos de dados em qualquer domínio de problema devam seguir essa regra; em outras palavras, pequenas quantidades de dados ausentes podem conter informações essenciais que devem ser gerenciadas (MCMAHON; ZHANG; DWIGHT, 2020). Para abordar essa questão, a literatura sugere o uso de métodos de Imputação Simples (IS) de dados ausentes, que envolvem a substituição de dados ausentes por valores reais (plausíveis). Embora essa abordagem permita reter mais dados em comparação com a exclusão, ela requer tempo para gerar valores de substituição razoáveis (FARHANGFAR; KURGAN; DY, 2008; REN et al., 2023), pode introduzir viés e subestimar a incerteza nos resultados (RUBIN, 1988; LI; STUART; ALLISON, 2015).

Para superar essa limitação, Rubin (2004) introduziu uma estratégia de imputação padrão-ouro na comunidade científica - Imputação Múltipla (IM) - para lidar com dados ausentes. Ao contrário das abordagens de IS, este método busca encontrar uma solução única na qual m soluções completas são criadas no banco de dados operacional, com $m > 1$. Essas soluções devem ser analisadas separadamente e combinadas para obter a melhor solução (LOBATO, 2016; NUNES; KLUCK; FACHEL, 2009). Para reduzir o erro na predição de valores ausentes, o uso de meta-heurísticas pode otimizar o valor a ser imputado (LOBATO, 2016). Notavelmente, estratégias bioinspiradas como algoritmos genéticos (AGs) são proeminentes na otimização de soluções (CHIU et al., 2022).

Os AGs foram propostos por Holland (1975). É uma meta-heurística de otimização

baseada na "sobrevivência do mais apto", inspirada na teoria evolucionária de Charles Darwin. Neste contexto, o algoritmo genético realiza uma busca propabilística baseado em mecanismos de seleção natural e genética. Maiores detalhes sobre a estrutura e funcionamento dos AGs serão tratados no Capítulo 2.

Em relação ao uso de AGs para Múltiplas Imputações, é crucial reconhecer o trabalho de Garcia, Kalenatic e Bello (2011) e o algoritmo MultImp (LOBATO, 2016). O algoritmo MultImp serve como a pedra angular desta pesquisa. Esse algoritmo empregou algoritmos genéticos para múltiplas imputações e também foi aplicado em cenários de Classificação Multirrótulo - *Multi-label Classification* (MLC). Garcia, Kalenatic e Bello (2011) e Lobato (2016) afirmam que tarefas de mineração de dados, especialmente aquelas relacionadas à classificação de dados, são sensíveis à abordagem de dados ausentes. Além disso, tarefas de classificação são amplamente utilizadas para avaliar a acurácia - *Accuracy* (ACC) de métodos de imputação (PROVOST; SAAR-TSECHANSKI, 2007; FARHANGFAR; KURGAN; DY, 2008; GARCIARENA; SANTANA, 2017).

Consequentemente, quanto maior a precisão na classificação, mais bem-sucedido é o método de imputação. No entanto, apenas alguns estudos empregaram MLC. Ao contrário da Classificação Monorrótulo *Single-Label Classification* (SLC), ou simplesmente classificação de dados, que associa um exemplo a um único rótulo, o MLC permite que uma instância seja associada a vários rótulos, aumentando assim a complexidade das tarefas de classificação (READ et al., 2011; GHANI; RAFI; TAHIR, 2020). Mais detalhes sobre esse tópico serão abordados no Capítulo 2.

Considerando a importância de lidar com valores ausentes na análise de dados e as soluções disponíveis na literatura existente, este trabalho apresenta uma abordagem algorítmica eficiente para imputações múltiplas aplicadas a tarefas de classificação multirrótulo. Este método é denominado EvoImp, uma combinação dos termos "evolucionário" e "imputação". Além disso, o nome é inspirado no MultImp (LOBATO, 2016), que serve como base para o algoritmo proposto nesse trabalho e se demonstrou promissor em suas fases iniciais para imputações múltiplas com dados ausentes.

O EvoImp aprimora a parametrização do MultImp para maximizar suas capacidades de imputação e explora novas configurações para experimentos computacionais. Nesse contexto, uma das caracterísicas mantidas do MultImp é trabalhar com um tamanho de população relativamente pequena e entende-se que essa é uma das abordagens utilizadas em soluções de *micro-genetic algorithms* (micro-GA) (COELLO; PULIDO, 2001; ABDI; ASADPOUR; SEYFARI, 2023). Uma vez que essa foi a única características relacionada ao micro-GA, optou-se por não denotar a solução proposta com essa abordagem.

Foi realizado um processo rigoroso de *benchmarking* para validar o desempenho do método proposto usando diversos conjuntos de dados de classificação multirrótulo. O EvoImp foi comparado com métodos de imputação estabelecidos e documentados na

literatura. Esses conjuntos de dados foram sistematicamente submetidos a seis taxas de valores ausentes para simular o mecanismo *Missing Completely At Random* (MCAR). Os resultados desses experimentos foram avaliados usando cinco classificadores distintos. Essa avaliação fornece *insights* sobre os pontos fortes e possíveis limitações do EvoImp quando aplicado a cenários de classificação multirrótulo do mundo real. Ao abordar os desafios associados aos dados ausentes nesse contexto, esse trabalho visa avançar na classificação multirrótulo e no campo mais amplo da análise de dados.

1.1 OBJETIVOS

O objetivo geral deste trabalho consiste em otimizar o tratamento de Valores Ausentes para imputação múltipla de dados em classificação multirrótulo.

1.1.1 Objetivos Específicos

- Realizar uma revisão da literatura sobre técnicas de imputação de dados ausentes, com foco em métodos baseados em algoritmos genéticos (AGs) e classificação multirrótulo;
- Investigar e propor o uso de algoritmos genéticos para realizar a imputação múltipla de dados ausentes em cenários de classificação multirrótulo;
- Avaliar a eficácia do algoritmo proposto em diferentes configurações de dados, taxas de valores ausentes e complexidade na classificação multirrótulo, considerando principalmente a acurácia do método em comparação com abordagens tradicionais de imputação.

1.1.2 Contribuições

Considerando a importância do tratamento dos valores ausentes no processo de análises de dados e as alternativas utilizadas na literatura para lidar com tal problemática, o presente trabalho investigará uma solução algorítmica eficiente para a imputação múltipla de dados aplicado em tarefas de classificação multirrótulo.

Do ponto de vista dos resultados, este trabalho tem como contribuições:

- Disponibilizar um método eficiente para o tratamento de valores ausentes, baseado em algoritmos genéticos, no cenário multirrótulo;
- Fornecer um material teórico e prático sobre o estudo dos valores ausentes, de métodos de imputação de dados, algoritmos genéticos e da classificação multirrótulo;

- Disponibilizar um software contendo um método eficiente para o tratamento de valores ausentes, baseado em algoritmos genéticos, no cenário multirrótulo, visando a replicabilidade dos estudos.

Vale destacar que essas contribuições foram depositadas no retpositório GitHub do projeto, o qual pode ser acessado pelo link: <https://github.com/jacobjr/EvoImp>.

1.2 ORGANIZAÇÃO DO DOCUMENTO

O restante deste trabalho está estruturado da seguinte forma:

Capítulo 2 - Fundamentação teórica: neste capítulo são apresentados os conceitos correlacionados com a pesquisa. De forma mais específica, disserta-se sobre mineração de dados e os processos de descoberta do conhecimento. Em seguida, apresentam-se os conceitos em relação a valores ausentes e os mecanismos que podem gerar a ausência de dados. Também são expostos conceitos associados aos métodos de imputação simples e múltipla de dados com destaque às técnicas utilizadas neste trabalho. Posteriormente, são apresentados os conceitos em relação à computação evolucionária, mais especificamente algoritmos genéticos e estratégias multiobjetivo. Por fim, é relatado o que são problemas de classificação multirrótulo fazendo uma comparação com rótulo único e abordam-se os classificadores multirrótulo utilizados nesse trabalho, bem como os métodos estatísticos para avaliação dos métodos de classificação.

Capítulo 3 - Trabalhos relacionados: neste capítulo os trabalhos correlacionados à esta tese são discutidos. Para melhor destacar os tipos de trabalho por tema, o capítulo encontra-se dividido em quatro seções. Primeiro são apresentadas as revisões de literatura encontradas no tema de valores ausentes, os quais demonstram que não se têm trabalhos envolvendo classificação multirrótulo e valores ausentes; em seguida, discutem-se trabalhos no tema de MLC e como esses realizam o tratamento de valores ausentes; posteriormente, são apresentados trabalhos que abordam o tratamento de valores ausentes na classificação multirrótulo, cujo foco é a resolução de rótulos faltantes (*missing label*); por fim, discutem-se os trabalhos que utilizam computação evolucionária no tratamento de VAs e faz-se uma sumarização dos pontos fortes e limitações das abordagens.

Capítulo 4 - EvoImp - Método proposto: este capítulo apresenta as etapas adotadas que constituem o método proposto nessa tese. Uma vez que o método é baseado em computação evolucionária, a estruturação das seções é feita a partir da estrutura de um Algoritmo Genético. Neste caso, primeiramente é explicado como foi

realizada a codificação dos indivíduos e a formação da população inicial. em seguida, é detalhado o funcionamento dos operadores genéticos utilizados. Posteriormente, é apresentada a função de aptidão do AG dando ênfase na explicação da abordagem de ordenação lexicográfica. Por fim, é feita a apresentação do algoritmo do EvoImp.

Capítulo 5 - Experimentos computacionais: neste capítulo são descritos os detalhes da configuração dos experimentos realizados. Inicialmente, é feito o detalhamento dos conjuntos de dados (*datasets*) escolhidos. Posteriormente, é realizada a descrição dos parâmetros relacionados aos valores ausentes, dos métodos de imputação e classificação, bem como os inerentes ao algoritmo genético. Em seguida, são indicados as tecnologias utilizadas no desenvolvimento do método computacional. Por fim, é apresentado como foi realizado o cálculo da complexidade computacional do EvoImp.

Capítulo 6 - Resultados e análises: esse capítulo trata sobre o desempenho do método. Primeiramente, são apresentados os resultados obtidos para cada um dos classificadores testados. Posteriormente, na discussão dos resultados é realizado um destaque no desempenho do EvoImp em comparação com técnicas de imputação de dados existentes.

Capítulo 7 - Conclusão e trabalhos futuros: apresenta as considerações finais sobre os resultados, destacando as contribuições da pesquisa. Em seguida, são apresentados alguns desafios traçados como trabalhos futuros a serem realizados. Por fim, são listadas as produções (bibligráfica e téncicas) geradas no contexto dessa tese.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os conceitos explorados para o desenvolvimento do estudo. De maneira mais específica, são discutidas a análise de dados e os procedimentos de descoberta de conhecimento. Em seguida, são delineados os conceitos relativos a VAs e os mecanismos que conduzem à sua ocorrência. Também são apresentados os conceitos ligados às técnicas de imputação de dados simples e múltipla, com ênfase nas metodologias empregadas nesta pesquisa. Posteriormente, são elucidados os conceitos referentes à computação evolutiva, especialmente os algoritmos genéticos e as abordagens multiobjetivo. Por último, são explorados os desafios da classificação de múltiplos rótulos, comparando-a com a classificação de rótulo único, e são discutidos os classificadores de múltiplos rótulos utilizados neste estudo, bem como os métodos estatísticos empregados na avaliação desses classificadores.

2.1 MINERAÇÃO DE DADOS

Vivencia-se uma explosão na geração de dados diários, impulsionada pela era digital. Cada interação *online*, transação e dispositivo conectado contribuem para essa proliferação. Dentre as áreas-chave que proporcionam esse fenômeno, destacam-se: *Big Data*, IoT e Mídias Sociais.

Big Data refere-se a conjuntos de dados massivos que desafiam capacidades tradicionais de processamento (WU et al., 2013). O paradigma da Internet das Coisas surge da proliferação de objetos e dispositivos móveis (ou "coisas") conectados, resultando na aquisição de fluxos periódicos de eventos de diferentes dispositivos e sensores que precisam ser processados (BEZERRA et al., 2021). As Mídias Sociais emergem como fonte de dados valiosa, revelando preferências, opiniões e comportamentos humanos (CIRQUEIRA et al., 2018). Plataformas como Facebook, Twitter e Instagram conectam bilhões de pessoas, formando uma tapeçaria social.

Com vastas quantidades de dados agora disponíveis, empresas em praticamente todas as indústrias estão concentradas em explorar dados para obter vantagem competitiva (TCHUENTE; HADDADI, 2023; KAMEL, 2023). O volume e a variedade de dados superaram amplamente a capacidade de análise manual e, em alguns casos, ultrapassaram a capacidade de bancos de dados convencionais. Ao mesmo tempo, os computadores tornaram-se significativamente mais poderosos, a rede está ubíqua, e algoritmos foram desenvolvidos para conectar conjuntos de dados, possibilitando análises mais amplas e profundas do que anteriormente possível (SEVILLA et al., 2022). A convergência desses

fenômenos deu origem à aplicação cada vez mais difundida da ciência de dados (PROVOST; FAWCETT, 2013; SHU; YE, 2023).

Nesse cenário, Rezende (2003) destaca que a análise inteligente de dados proporciona benefícios significativos para a entidade ou indivíduo, viabilizando a obtenção de informações ao analisar e contextualizar dados. Adicionalmente, esse processo possibilita a criação de conhecimento, resultante da comparação e combinação de informações úteis e relevantes.

Em resumo, Han, Kamber e Pei (2012) descrevem a abundância de dados, aliada à necessidade de ferramentas poderosas de análise de dados, como uma situação em que há muitos dados, mas pouca informação. Nesse caso, é crucial empregar procedimentos específicos e recursos computacionais para obter uma compreensão completa dessas informações, o que caracteriza o processo conhecido como KDD (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996; HAN; KAMBER; PEI, 2012; SHU; YE, 2023).

Fayyad, Piatetsky-Shapiro e Smyth (1996) descrevem o KDD como um processo complexo, composto por várias etapas que são interativas e iterativas. Esse processo tem como objetivo identificar padrões compreensíveis, válidos, inovadores e potencialmente úteis a partir de conjuntos de dados. Uma medotologia de processo que vem se destacando na área é a *Cross Industry Standard Process for Data Mining* (CRISP-DM) (WIRTH; HIPP, 2000; BOKRANTZ; SUBRAMANIYAN; SKOOGH, 2023).

A CRISP-DM é aplicada por meio de um processo hierárquico, que é composto por um conjunto de tarefas descrevendo quatro níveis de abstração (VANEGAS et al., 2023). Em consonância com esses níveis de abstração, os estudos conduzidos por Sousa et al. (2020), BRZOZOWSKA et al. (2023), Krishnaswamy et al. (2023) apresentam estudos de casos da utilização dos componentes do CRISP-DM.

2.2 VALORES AUSENTES

Little e Rubin (2019) apresentam que termos como valores ausentes e dados incompletos são frequentemente utilizados para descrever lacunas em conjuntos de dados. O autor Longford (2005) descreve a ausência de um dado em termos de uma medida que não pode ser obtida. No entanto, ao traduzir esses conceitos para o português, há uma falta de consenso, resultando em diversas denominações, como dados faltosos, dados faltantes e dados incompletos, entre outras (BLEIDORN et al., 2022; ALMEIDA et al., 2023; FREITAS, 2022). Segundo Finatto, Silva e Esteves (2021), a falta de consenso e diversidade é apontada pela área de Linguística Aplicada ou Linguística Descritiva como um fenômeno na comunicação técnico-científica. Essa fato ocorre quando o mesmo termo é utilizado por diferentes interlocutores e/ou está presente em diversas áreas de pesquisa, o que reforça a diversidade do tema. Lobato (2016) estende este conceito ao diferenciar as instâncias de uma base de dados entre casos completos e casos incompletos, no qual, como

a própria denominação sugere, são formadas por instâncias que apresentam ou não dados faltosos.

Base de dados do mundo real apresentam uma quantidade significativa de dados ausentes. Neste contexto, a verificação da extensão destes valores ausentes é um dos passos iniciais no processo de análise de dados (SAINANI, 2015). A Tabela 1 exemplifica o conceito de dados ausentes. Os quatro primeiros registros representam casos completos, onde todas as informações sobre alguns carros estão disponíveis. Por exemplo, tem-se carros de anos específicos (2019, 2020, 2018, 2021) com dados completos sobre a potência em cavalos, o tipo de combustível (gás, híbrido, elétrico) e o tipo de transmissão (automática, continuamente variável, manual). No entanto, os quatro últimos registros ilustram casos incompletos, nos quais algumas informações estão ausentes e representadas pelo símbolo "?". Essa ausência de dados pode ocorrer por várias razões, como falhas na coleta de informações, erros durante a entrada de dados ou até mesmo porque certos atributos podem não ser aplicáveis a todos os carros.

Atr.1	**Atr.2**	**Atr.3**	**Atr.4**
2019	200	Híbrido	Automática
2020	180	Elétrico	Cont. Variável
2018	150	Gás	Manual
2021	220	Híbrido	Automática
?	190	Gás	Manual
2022	?	Híbrido	Automática
2017	170	?	Cont. Variável
2019	205	Elétrico	?

Tabela 1 – Tabela de exemplo de dados ausentes.

Essa representação de dados ausentes destaca a importância de lidar com a incompletude de informações em conjuntos de dados, uma vez que a presença de valores faltantes pode impactar análises estatísticas e modelos preditivos (LUENGO; GARCÍA; HERRERA, 2012; LITTLE; RUBIN, 2019). Neste contexto, uma base de dados pode conter valores ausentes em diferentes instâncias e atributos, sendo necessária estudar o que pode ter causado estas faltas a fim de direcionar o tratamento.

2.2.1 Mecanismos de Ausência de Dados

Diversas razões podem causar a ausência de dados em base reais. A identificação deste padrão de ausência é um dos principais aspectos para direcionar os métodos para lidar com estas ausências (GARCIARENA; SANTANA, 2017). Neste contexto, destacamos os três principais mecanismos (LOBATO, 2016; GARCIARENA; SANTANA, 2017):

- *Missing Completely At Random* (MCAR): em português ausência completamente aleatória. Caso no qual a probabilidade do atributo faltoso é independente de qualquer

influência detectável (tanto valores ausentes quanto observados) e não depende dos valores de entrada;

- *Missing At Random* (MAR): em português ausência aleatória. Ocorre quando a ausência de dados é independente dos valores faltosos, porém um padrão desta falta pode ser predito a partir de outros atributos da base de dados;
- *Missing Not At Random* (MNAR): em português ausência não aleatória. Este caso é similar ao MAR, porém, neste caso, os valores que causam a falta de outros não são conhecidos. Este tipo pode ter duas origens:
  - *Missingness depending on unobserved Variables*: em português falta depende de variáveis não observadas. Uma das causas deste valor ser desconhecido é devido este simplesmente não ter sido observado;
  - *Missingness depending on its Value Itself*: em português falta depende do seu próprio valor. Um atributo pode faltar dependendo do seu valor ou de uma certa faixa de valores (limites).

Dada a característica dos mecanismos de ausência e o processo de validação a ser explicado, esta tese optou por utilizar conjuntos de dados completos e induzir os valores ausentes, processo conhecido na literatura como amputação dos dados (SCHOUTEN; LUGTIG; VINK, 2018).

2.3 MÉTODOS DE IMPUTAÇÃO DE DADOS

Para enfrentar o desafio de dados faltosos, a literatura sugere a aplicação de técnicas de imputação de dados, as quais consistem na substituição de valores faltantes por dados reais e plausíveis. Embora essa estratégia permita preservar mais informações em comparação com a exclusão de registros incompletos, é importante mencionar que a geração de valores de substituição adequados pode demandar tempo (FARHANGFAR; KURGAN; DY, 2008; REN et al., 2023).

Conforme mencionado no capítulo 1, uma abordagem simples, porém muitas vezes eficaz, para enfrentar o problema de dados ausentes é a Imputação Simples. Nesse método, os valores faltantes são preenchidos com um único valor estimado, frequentemente derivado da média, mediana ou por meio de modelos de regressão (LIN; TSAI, 2020). É crucial observar que, embora a Imputação Simples seja uma solução rápida, ela pode não capturar completamente a complexidade subjacente dos dados, especialmente em situações em que a variabilidade é significativa. Portanto, a escolha do método de imputação deve ser cuidadosamente ponderada de acordo com a natureza dos dados e os objetivos da análise (LUENGO; GARCÍA; HERRERA, 2012).

Para contornar essa limitação, uma abordagem pioneira foi introduzida por Rubin (2004), conhecida como Imputação Múltipla. Essa estratégia representa um padrão-ouro no tratamento de dados ausentes. Ao contrário das técnicas de Imputação Simples, a IM propõe a criação de múltiplas soluções completas da base de dados, onde m conjuntos de dados completos são gerados, sendo $m > 1$. Cada uma dessas soluções deve ser analisada de forma independente, e posteriormente, os resultados são combinados para produzir a solução final (LOBATO, 2016; NUNES; KLUCK; FACHEL, 2009).

Dessa forma, seguindo as diretrizes do trabalho de Luengo, García e Herrera (2012) destaca-se os seguintes métodos de imputação, os quais foram utilizados nesta tese:

- **KNNI:** Sempre que houver um valor ausente, os K-vizinhos mais próximos à instância que contém o VA são determinados. O valor mais comum entre os K-vizinhos mais próximos foi usado para imputar atributos nominais. Para atributos numéricos, a imputação é realizada calculando a média dos valores vizinhos (BATISTA; MONARD et al., 2002);
- **WKNNI:** Essa técnica envolve a determinação das distâncias entre os K-vizinhos mais próximos e uma distribuição de ponderação em relação às distâncias entre cada vizinho. Após isso, o processo KNNI foi repetido (LING; DONG-MEI, 2009);
- **KMI:** Essa técnica divide um banco de dados em clusters com base em suas características. Uma vez feito isso, a técnica dos K-vizinhos mais próximos é aplicada ao decidir qual valor deve ser imputado (HRUSCHKA; HRUSCHKA; EBECKEN, 2005);
- **MC:** Neste método, o valor mais comum é adotado para imputação em atributos nominais e a média de todos os atributos correspondentes no caso de atributos numéricos (GRZYMALA-BUSSE; HU, 2001);
- **CMC:** Este método faz o mesmo que MC, mas apenas utiliza a classe de atributo referenciada com VA (GRZYMALA-BUSSE; HU, 2001).

2.4 algoritmos genéticos

Os algoritmos genéticos se destacam por proporcionar a resolução de problemas complexos de forma rápida e confiável. Enquanto as técnicas tradicionais podem falhar na otimização e busca de problemas, os AGs se sobressaem (FREITAS, 2002). Essas técnicas tradicionais geralmente começam com um único candidato (indivíduo) e aplicam heurísticas, muitas vezes estáticas e diretamente ligadas ao problema em questão. Por outro lado, os AGs operam sobre uma população de candidatos (vários indivíduos) de forma paralela, realizando operações em diferentes regiões do espaço de solução.

O funcionamento básico de um AG é apresentado no Algoritmo 1. Esse algoritmo recebe como entrada um conjunto de parâmetros que guiam sua execução e retorna uma solução ótima ou aproximada.

Algoritmo 1: Algoritmo genético canônico baseado em Lobato (2016), Kumar et al. (2010).

```
   Input: Conjunto de Parâmetros
   Output: Solução
 1 Inicializar a População Inicial;
 2 while Critério de parada não alcançado do
 3 |   Avaliar a função de aptidão;
 4 |   while tamPopAtual < Número de indivíduos da nova geração do
 5 |   |   Selecionar indivíduos para cruzamento;
 6 |   |   Aplicar Crossover;
 7 |   |   Aplicar mutação;
 8 |   |   Adicionar o Indivíduo à População Atual: PopAtual ← individuo;
 9 |   end
10 |   Ordenar PopAtual;
11 end
12 return bestIndividual;
```

O algoritmo começa inicializando a população inicial de indivíduos, que representam possíveis soluções para o problema em questão (linha 1). Em seguida, entra em um *loop* principal, onde avalia a função de aptidão de cada indivíduo na população atual (linha 2-3). Enquanto o critério de parada não é alcançado, o algoritmo continua a evoluir a população.

Em seguinda, o operador genético de reprodução é acionado (linhas 5-6). Esse operador é responsável por gerar as novas populações. Isso ocorre através da seleção de dois pais (linha 5), cujos cromossomos (estruturas de dados que representam possíveis soluções do problema) são divididos em uma posição escolhida aleatoriamente. Os segmentos resultantes de cada pai são então combinados para gerar novos cromossomos (linha 6) (REZENDE, 2003). Para garantir a qualidade da nova geração da população, os cromossomos são selecionados com base em uma função que identifica os de maior aptidão para o problema proposto (KUMAR et al., 2010). Entre essas funções, a Normalização Linear seleciona os cromossomos com base em sua aptidão; ou seja, quanto maior a aptidão, maior a probabilidade de seleção. Outra função de seleção é realizada através de torneios, nos quais dois indivíduos são selecionados aleatoriamente para determinar o melhor indivíduo.

Além disso, a mutação é aplicada em alguns indivíduos para introduzir variação genética na população (linha 7). A mutação consiste em uma alteração aleatória (normalmente com baixa probabilidade) de cada gene do cromossomo. Essas perturbações nas cadeias dos cromossomos garantem que a busca não fique estagnada em sub-regiões do espaço de busca, permitindo que qualquer ponto do espaço seja alcançado (REZENDE, 2003).

Após a formação da nova geração, a população atual é atualizada com os novos indivíduos gerados (linha 8). Esta população é então ordenada de acordo com o desempenho de seus membros, para que os melhores indivíduos fiquem no topo (linha 10). Essa ordenação é realizada pelo operador genético de elitismo, o qual, conforme Kumar et al. (2010), consiste em preservar imediatamente os melhores elementos da nova geração da população para evitar a perda das melhores soluções de uma geração para outra.

O processo iterativo continua até que o critério de parada seja atingido, como um número máximo de gerações ou a convergência da solução. Finalmente, o algoritmo retorna o melhor indivíduo encontrado durante as iterações, que representa a solução ótima ou aproximada para o problema (linha 12).

2.5 CLASSIFICAÇÃO MULTIRRÓTULO

Em problemas de classificação de rótulo único, um conjunto de rótulos de classe é predeterminado, e cada objeto deve ser associado a um e apenas um rótulo (GONÇALVES; FREITAS; PLASTINO, 2018). Formalmente, seja X o espaço de entrada/características, e y denote o valor da classe, onde $y \in L$, que é o espaço de saída (um conjunto de rótulos de classe disjuntos). Neste caso, cada amostra está estritamente associada a um único rótulo de classe (NGUYEN et al., 2019; SÁ et al., 2020).

No entanto, há cada vez mais contextos nos quais os dados podem pertencer a mais de um rótulo de classe. Esta condição de classificação é referida como classificação multirrótulo. Inicialmente, a MLC focava principalmente em tarefas como categorização de texto, classificação de função de proteínas, categorização de música, classificação semântica de cenas e diagnóstico médico (GONÇALVES; FREITAS; PLASTINO, 2018; NGUYEN et al., 2019; VENKATESAN; ER, 2014).

Recentemente, novas aplicações surgiram em Visão Computacional, Processamento de Linguagem Natural e Mineração de Dados, incluindo Anotação de Vídeo, Mineração de Texto Legal e Perfil de Usuário (LIU et al., 2021).

De acordo com Tsoumakas, Katakis e Vlahavas (2010) e Sá et al. (2020), semelhante à SLC, a MLC é representada por X e y, onde cada amostra $x \in X$ é atribuída a um subconjunto do espaço de saída (um conjunto de rótulos de classe não disjuntos). A Tabela 2 ilustra um exemplo fictício que mostra a diferença entre SLC e MLC, adaptado de Tang, Rajan e Narayanan (2009). Considerando que os dados na Tabela 2 compreendem 5 instâncias $(x_1, x_2, x_3, x_4, x_5)$ e 3 rótulos (y_1, y_2, y_3).

A Tabela 2a ilustra o cenário de SLC, onde cinco instâncias de dados (x_1 a x_5) estão estritamente associadas a um único rótulo (y_1 a y_3). Por exemplo, x_1 está associado a y_1, x_2 está associado a y_2, e assim por diante. Por outro lado, a MLC permite que instâncias

Tabela 2 – Comparação entre SLC e MLC usando um exemplo ilustrativo com 5 instâncias e 3 rótulos.

Dados	**Rótulo**
x_1	y_1
x_2	y_2
x_3	y_3
x_4	y_1
x_5	y_3

(a) Rótulo Único

Dados	**Rótulos**
x_1	y_1, y_2
x_2	y_2, y_3
x_3	y_1, y_3
x_4	y_2
x_5	y_3

(b) multirrótulo

de dados sejam associadas a vários rótulos simultaneamente. A Tabela 2b demonstra o cenário de MLC, onde as mesmas cinco instâncias de dados (x_1 a x_5) podem ter vários rótulos atribuídos a elas. Por exemplo, x_1 está associado tanto a y_1 quanto a y_2, x_2 está associado tanto a y_2 quanto a y_3, e assim por diante. Essa distinção destaca como a SLC restringe cada instância de dados a um único rótulo, enquanto a MLC permite que as instâncias pertençam a vários rótulos simultaneamente, tornando-a mais adequada para cenários em que objetos ou pontos de dados podem ser associados a diferentes classes.

Embora a diferença seja sutil na teoria, a MLC tende a ser mais desafiadora na prática. Os autores Gonçalves, Freitas e Plastino (2018) e Sá et al. (2020) enumeraram as seguintes razões para isso:

- As possíveis classes de uma determinada instância (espaço de saída) em MLC crescem exponencialmente com o aumento do número de rótulos. Portanto, ao considerar que um problema possui L rótulos distintos, o tamanho do espaço de saída em MLC é 2^L (combinação de rótulos), enquanto é apenas L em SLC;
- Um algoritmo MLC deve considerar se existe ou não uma correlação entre rótulos. Esse tipo de correlação é uma etapa essencial para garantir a eficácia de vários processos MLC (NGUYEN et al., 2019; QIAN et al., 2022; SUN et al., 2021);
- A avaliação de desempenho de sistemas MLC utiliza métricas diferentes das tradicionalmente usadas em SLC (GIBAJA; VENTURA, 2015). Em SLC, a avaliação de uma nova instância pode estar correta ou incorreta. Por outro lado, em MLC, o resultado pode ser parcialmente correto. Isso ocorre quando o classificador prevê alguns rótulos corretos, mas inclui algumas previsões incorretas ou até mesmo omite um rótulo que deveria ser previsto. Esse problema requer atenção cautelosa, pois algumas métricas seguem aspectos contrastantes para definir o que é uma boa previsão MLC (SÁ et al., 2020; PEREIRA et al., 2018);
- Ao contrário dos problemas SLC, que tradicionalmente envolvem a análise de dados relacionais (estruturados), as aplicações MLC normalmente lidam com tarefas de *big*

data, que envolvem dados semi-estruturados ou não estruturados (NGUYEN et al., 2019; ZHENG et al., 2019).

Todos esses desafios ampliaram a complexidade associada ao tratamento de VAs. No entanto, encontrar estudos que relacionem MLC e VAs não é simples, como demonstrado em Emmanuel et al. (2021), Lin e Tsai (2020) e Chiu et al. (2022).

2.5.1 Classificadores Multirrótulo

A utilização de tarefas de classificação vem sendo bastante utilizada, na literatura, para mensurar a acurácia dos métodos de imputação, isto é, quanto maior for a acurácia da classificação, melhor o desempenho do método (GARCIARENA; SANTANA, 2017). Neste contexto, para o aprendizado multirrótulo desse trabalho foram utilizados os seguintes métodos de classificação, os quais serão detalhados a seguir: *Binary Relevance* (BR), *Hierarchy Of Multilabel classifiER* (HOMER), *Multi-Label k Nearest Neighbors* (MLKNN), *Classifier Chains* (CC) e *Ensemble of Classifier Chains* (ECC) (READ et al., 2011; TSOUMAKAS; KATAKIS; VLAHAVAS, 2008; TSOUMAKAS; KATAKIS; VLAHAVAS, 2010).

Binary Relevance

Uma das abordagens utilizadas para resolver problemas de classificação multirrótulo é por meio da transformação do problema em um ou mais problemas monorrótulo. Neste contexto, o *Binary Relevance* se destaque como o método mais comum para esta forma de transformação (TSOUMAKAS; KATAKIS, 2007).

Neste sentido, o BR realiza a transformação do problema multirrótulo em múltiplos problemas binários, isto é, realiza a criação um problema para cada uma das classes e cada modelo binário é treinado para realizar a predição de uma classe (READ et al., 2011). A Figura 1 ilustra um exemplo fictício que demonstra a utilização do BR, adaptado de Tang, Rajan e Narayanan (2009). Nesse exemplo, considera-se que os dados contém 4 instâncias $(1, 2, 3, 4)$, os quais podem ter até 4 rótulos (L_1, L_2, L_3, L_4).

Hierarchy Of Multilabel classifiER

O HOMER estabelece uma hierarquia de classificadores multirrótulo por meio de um algoritmo que segue o paradigma de divisão-e-conquista (SMITH, 1985). Neste caso, cada classificador trata com um pequeno número de classes (comparado com a quantidade do problema) (TSOUMAKAS; KATAKIS; VLAHAVAS, 2008).

Dentre os principais processos realizados pelo HOMER está a distribuição dos conjuntos de classe por meio de subconjuntos, os quais são agrupados por similaridade.

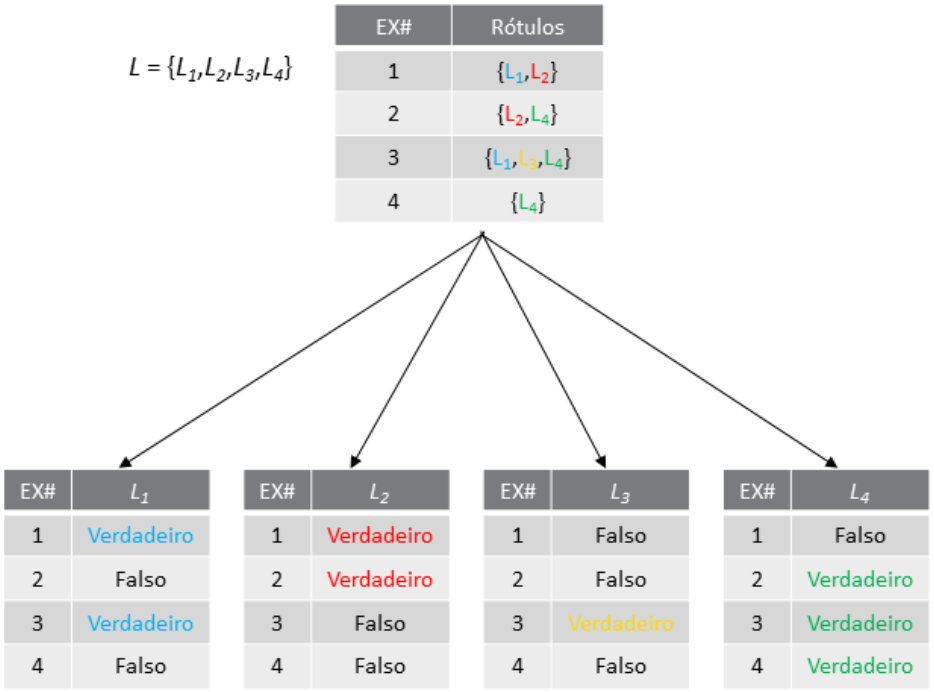


Figura 1 – Exemplo de funcionamento do BR. Fonte: Adaptado de Vidulin (2013)

Este processo é conhecido como *balanced clustering* (BANERJEE; GHOSH, 2006) e um exemplo de como é realizado pode ser observado na Figura 2.

O exemplo apresentado na Figura 2 contém oito possíveis rótulos $\{L_1, L_2, ..., L_8\}$. Nesse caso, o HOMER gera diferentes nós n de classificadores (h_n), os quais são responsáveis por classificar um certo subconjuntos de rótulos (L_i).

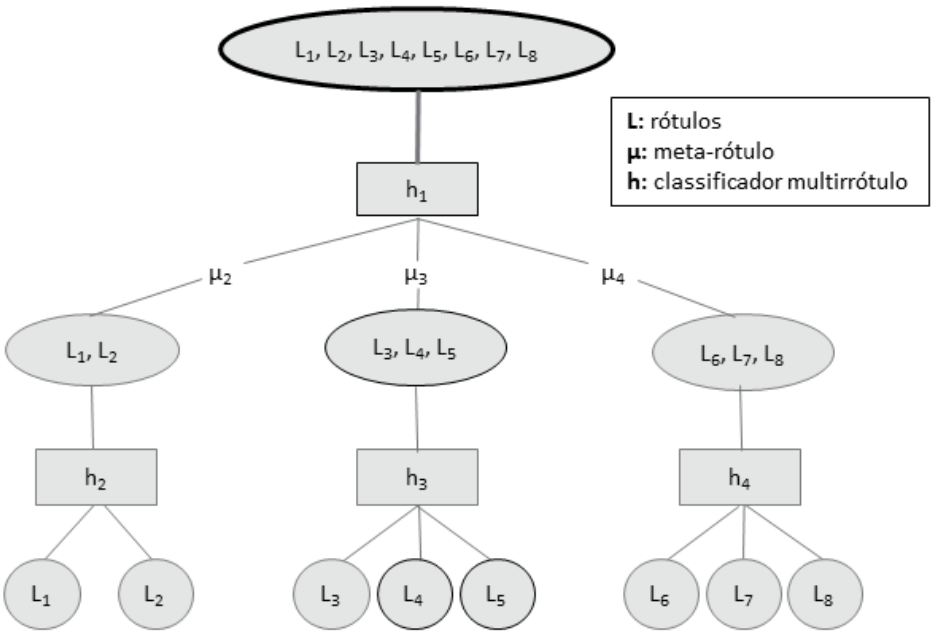


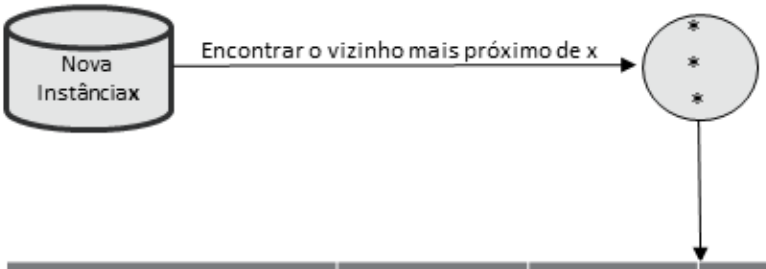
Figura 2 – Exemplo de funcionamento do *balanced clustering*. Fonte: Adaptado de Tsoumakas, Katakis e Vlahavas (2008).

Para realizar a disjunção dos rótulos, Tsoumakas, Katakis e Vlahavas (2008) definiu o conceito de *meta-rótulo* (μ_n), o qual delimita que um certo exemplo pode ser considerado pelo (μ_n), se ele estiver associado a pelo menos um dos rótulos associados ao *meta-rótulo* (μ_n).

Multi-Label k Nearest Neighbors

O ML-kNN é uma derivação do popular algoritmo *k-nearest neighbor* (kNN) (AHA, 1997). Nesta abordagem, o algoritmo, primeiramente, identifica os k vizinhos mais próximos da instância de teste. Com isso, uma lista de candidatos das instâncias vizinhas é obtida. Posteriormente, é empregado o princípio maximum a posteriori (MAP), a fim de predizer o conjunto de rótulos da instância de testes (ZHANG; ZHOU, 2007).

A Figura 3 demonstra o funcionamento do algoritmo por meio do tratamento de uma nova instância x, a qual pode ser atribuída com os rótulos $\{L_1, L_2, L_3, L_4\}$.



Rótulos	L1	L2	L3	L4
1º Vizinho	1	0	1	0
2º Vizinho	0	0	1	1
3º Vizinho	1	1	0	0
Probabilidades	P(l=1) = 0,66	P(l=1) = 0,33	P(l=1) = 0,66	P(i=1) = 0,33
	P(l=0) = 0,33	P(l=0) = 0,66	P(l=0) = 0,33	P(l=0) = 0,66
MAP Probabilidades (threshold = 0,5)	0,6	0,6	0,6	0,6
Rótulo final de x	1	0	1	0

Figura 3 – Exemplo de funcionamento do ML-kNN. Fonte: Adaptado de Settouti et al. (2019).

Para essa nova instância x são verificados 3 vizinhos e mapeado os rótulos vinculados a cada um desses. A partir desse mapeamento é calculada a probabilidade de cada rótulo da amostragem. Nesse caso, as probabilidades que estão acima de um *threshold* (no exemplo

$> 0,5$), é vinculado o rótulo a nova instância. Com isso, a nova instância x deverá receber os rótulos L_1 e L_3.

Classifier Chains

O modelo *Classifier Chains*, assim como o *Binary Relevance*, divide o problema de classificação multirrótulo em $|L|$ classificadores binários H. Os classificadores formam uma cadeia de classificadores binários $\{h_1, h_2, ..., h_{|L|}\}$, sendo que cada classificador propaga as informações entre os classificadores (READ et al., 2011). Essas características de propagação fazem com que o modelo leve em consideração a correlação entre as classes, superando uma das limitações ocasionadas no método BR (READ et al., 2009).

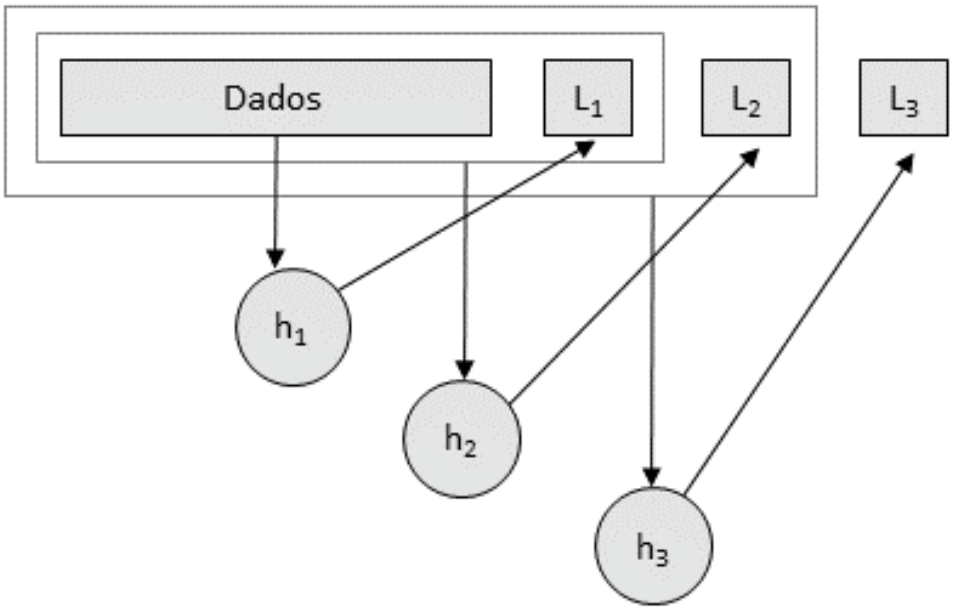


Figura 4 – Exemplo de funcionamento do CC. Fonte: Adaptado de Riemenschneider et al. (2017).

O exemplo da Figura 4 ilustra a classificação de uma nova instância (*Dados*) e uma cadeia com três classificadores (h_1, h_2 e h_3). Cada um desses classificadores está responsável por um rótulo (represantados por L_i) e propaga a informação em cadeia para os classificadores seguintes.

Ensemble of Classifier Chains

Um dos problemas que pode ocorrer no algoritmo de *Classifier Chain* é a cadeia de classificadores estar precariamente ordenada e, por consequência, gerar um possível efeito de propagação do error. Nesse sentido, Read et al. (2011) propôs um conjunto de cadeia, os quais tem os rótulos ordenados randomicamente. Essa estratégia consegue reduzir de maneira globla o risco de propagação negativa na classificação.

2.5.2 Medidas de Desempenho

Conforme discutido anteriormente, tarefas de classificação são amplamente utilizadas para avaliar métodos de imputação (FARHANGFAR; KURGAN; DY, 2008; PROVOST; SAAR-TSECHANSKI, 2007; GARCIARENA; SANTANA, 2017). Geralmente, o procedimento adotado envolve o cálculo da estatística para as instâncias completas e, em seguida, para o conjunto de dados imputado. A diferença entre esses cálculos é então computada, buscando reduzir essa discrepância, também conhecida como erro associado.

No contexto da classificação multirrótulo, a literatura destaca que informações relacionadas à construção do modelo são consideradas métricas mais preferíveis para avaliar o impacto da imputação (TSOUMAKAS; KATAKIS; VLAHAVAS, 2008; GONÇALVES; FREITAS; PLASTINO, 2018).

A notação usada por Gonçalves, Plastino e Freitas (2013) e Lobato (2016) foram utilizadas para descrever as medidas adotadas nesse trabalho: (i) n: número de instâncias no conjunto de teste; (ii) q: número de rótulos; (iii) Y_i: conjunto de rótulos originais, por exemplo, i; e (iv) Z_i: conjunto de rótulos preditivos, por exemplo, i.

- ***Exact Match*** calcula, usando um sistema binário, se todos os rótulos da instância são previstos corretamente. Esta medida, conforme expresso na Eq. 2.1, é considerada trivial porque ignora previsões parciais:

$$EM = \frac{1}{n}\sum_{i=1}^{n} I(Y_i = Z_i) \tag{2.1}$$

- ***Acurácia*** é também uma medida que conta os rótulos previstos corretamente de uma instância. Neste caso, previsões parciais são levadas em consideração. A Eq. 2.2 expressa o modelo matemático dessa medida:

$$ACC = \frac{1}{n}\sum_{i=1}^{n} \frac{Y_i \cap Z_i}{Y_i \cup Z_i} \tag{2.2}$$

- ***Hamming Loss*** é uma medida que, ao contrário da *acurácia*, avalia o desempenho do classificador encontrando a média de previsões incorretas. A Eq. 2.3 descreve essa medida:

$$HL = \frac{1}{n}\sum_{i=1}^{n} \frac{Y_i \Delta Z_i}{q} \tag{2.3}$$

O Algoritmo Genético proposto nesta tese levou em consideração uma abordagem multi-objetivo utilizando ordem lexicográfica. Maiores detalhes sobre esse tema serão apresentados na Seção 4.3.

3 TRABALHOS RELACIONADOS

Neste capítulo, os estudos relacionados a esta tese são analisados. Para destacar de maneira mais eficaz os diversos tipos de pesquisa por tema, o capítulo está fragmentado em quatro partes. Inicialmente, são exploradas as revisões de literatura relacionadas ao tópico de VAs, revelando a ausência de estudos que abordem classificação múltirrótulos e valores ausentes. Em seguida, são examinadas as pesquisas sobre MLC e sua abordagem em relação aos dados faltantes. Posteriormente, são expostas as investigações que tratam da manipulação de valores ausentes na classificação de múltiplos rótulos, com um foco específico na resolução de *Missing Labels*. Por fim, são discutidos os estudos que aplicam a computação evolucionária para o tratamento de VAs, além de uma síntese das principais abordagens e suas respectivas limitações.

3.1 REVISÕES DE LITERATURA SOBRE VA

Encontrar trabalhos que relacionem MLC e VAs não é uma tarefa trivial, conforme pode ser constatado nas revisões sistemáticas e bibliométricas apresentadas nessa Seção. Primeiramente, o trabalho de Lin e Tsai (2020) realizou na análise sistemática da literatura em imputação de dados ausentes. Foram selecionados 111 trabalhos publicados no período de 2006 a 2017. Dentre os principais achados desse trabalho destaca-se: as principais fontes de *datasets*; quais as taxas de ausência mais encontradas; os mecanismos causadores de ausência de dados; as técnicas e abordagens de imputação predominantes; e as métricas de avaliação utilizadas nos trabalhos estudados. Nesse caso, vale destacar que nenhum dos trabalhos selecionados por Lin e Tsai (2020) eram no contexto de *Multi-label classification*.

Outra revisão sistemática no tema de valores ausentes é o de Chiu et al. (2022). Essa revisão focou em trabalhos que utilizaram algoritmos meta-heurísticos inspirados na natureza. O estudo delineou o perído de 2011 a 2021 e teve como achado 48 artigos no tema. Os achados principais corroboram com os encontrados em Lin e Tsai (2020) com destaque na análise específica das meta-herísticas bio-inspiradas e padronização dos trabalhos com experimentação utilizando poucas bases de dados. Novamente, nenhum dos artigos investigados trabalhou com problemas envolvendo MLC.

Por sua vez, os autores Nugroho e Surendro (2024) realizaram uma análise bibliométrica de artigos de imputação de valores ausentes no período de 2012 a 2023. Nesse caso, a análise levou em conta 352 artigos encontrados. Com relação aos resultados obtidos, pode-se destacar a descoberta dos temas prevalecentes na área culminando com a apresentação de uma mapa temático com a categorização dos principais conceitos utilizados no tema, bem como os assuntos que estão em ascensão e declínio nas pesquisas. Mais uma

vez, nenhum dos trabalhos estudados abordou problemáticas abrangendo MLC.

Como pode ser observado, exite uma lacuna quanto ao estudo de impacto de valores ausentes na classificação multirrótulo. Dentre as hipóteses de não se encontrar trabalhos nessa área é que os problemas de MLC aumentam a complexidade das tarefas de classificação e, também, a pouca disponibilização de *datasets* de qualidade (READ et al., 2011; SONG et al., 2023).

3.2 EXEMPLOS DE MLC

O trabalho de Nakayama et al. (2024) realizou a construção do *Brazilian Multilabel Ophthalmological Dataset of Retina Fundus Photos* (BRSET), em resposta à escassez de conjuntos de dados oftalmológicos diversificados nessas áreas, especialmente no Brasil e na América Latina e, também, como uma forma de mitigar vieses na pesquisa de Inteligência Artificial (IA) médica. Composto por 16.266 fotos coloridas do fundo da retina de 8.524 pacientes brasileiros, o BRSET integra informações sociodemográficas. O conjunto de dados não possui dados duplicados, e apenas 3 atributos contêm dados ausentes. Os atributos com dados ausentes apresentavam uma alta taxa de ausência (acima de 50%) e, nesse caso, somente foram removidos do *dataset*.

A fim de mostrar a diversidade de áreas que apresentam base de dados com VAs, destaca-se o artigo de Sharma et al. (2023), o qual investiga o desempenho de modelos de Aprendizagem de Máquina (AM) para classificação multirrótulo de dados de detecção de intrusão no contexto de rede da Internet das Coisas (*Internet of Things* - IoT). Os testes foram realizados ao mesclar dois conjuntos de dados de detecção de intrusão disponíveis em repositórios públicos e avaliar o desempenho dos modelos utilizando as métricas de avaliação: *precision*, *recall* e *F1-score*. Com relação a quantidade de dados ausentes, os autores não informam a taxa de ausência. No caso do tratamento dos VAs, para os atributos numéricos foi realizado o preenchimento manual pela média ou pelo valor mais provável, não ficando claro se realizaram a eliminação das instâncias com variáveis categóricas ou ordinais.

3.3 MLC E *MISSING LABELS*

Dentre as abordagens de tratamento de VAs na classificação multirrótulo, foram encontradas pesquisas relacionadas a tratativas de *Missing Labels* (ML) (WANG; LIN; LIU, 2019; CHENG; SONG; QIAN, 2021), o que significa focar na previsão de um ou mais rótulos desconhecidos. Pesquisas nessa área tem aumentado, devido a necessidade de atribuir multirrótulos em *datasets* inteiros, os quais inicialmente não tem essa informação ou está incompleta (HAN et al., 2023).

O trabalho de Wang, Lin e Liu (2019) apresenta uma seleção de características multirrótulo que considera a interação de características. Para isso, os autores usam as definições de entropia de informação de vizinhança multirrótulo e informação mútua de vizinhança multirrótulo para mitigar o impacto negativo de rótulos ausentes.

Abordagens que consideram correlações entre rótulos vêm se mostrando eficazes na recuperação de rótulos ausentes. No entanto, muitas vezes, essas abordagens são instáveis devido ao tratamento semelhante de rótulos positivos e negativos em desequilíbrio. Cheng, Song e Qian (2021) concentram-se em lidar com rótulos ausentes aproveitando correlações de rótulos e implementando um *autoencoder* de máquina extrema de núcleo em dois níveis. Os autores verificaram o método proposto em conjuntos de dados com rótulos ausentes e completos.

Uma abordagem semelhente ao de Cheng, Song e Qian (2021) é realizada por Qian et al. (2023). Nesse trabalho, o autor propõe a aprendizagem multirrótulo auto-dependente com um algoritmo de recuperação de rótulos duplos k. Nesse caso, são construídas duas matrizes de contagem de rótulos a partir da matriz de rótulos original, considerando os rótulos positivos e negativos de forma independente. Isso é feito por meio de estatísticas nos k vizinhos mais próximos de acordo com as características de entrada. Em seguida, as matrizes de rótulos positivos e negativos são decompostas e recuperadas usando a fatorização de matriz, nomeadamente o duplo k (k vizinhos mais próximos e k semânticas latentes).

3.4 algoritmos genéticos E VA

Como pode ser observado, um número limitado de estudos vêm abordando especificamente a questão de rótulos ausentes. Dado esse cenário, passou-se a buscar trabalhos que relacionem algoritmos genéticos no tratamento de dados ausentes. Tran, Zhang e Andreae (2015) propuseram um método de imputação de dados por meio de uma abordagem baseada em programação genética chamada GPMI. Uma estratégia de Imputação Múltipla foi aplicada neste método, e uma estimativa de valores ausentes foi realizada usando técnicas de regressão. O GPMI foi comparado com sete métodos de imputação por meio de um experimento realizado em oito conjuntos de dados e aplicando seis diferentes taxas de valores ausentes (5%, 10%, 20%, 30%, 40% e 50%) com o auxílio do MCAR como mecanismo de dados ausentes. A acurácia do classificador foi a medida de desempenho adotada. Os resultados sugerem que o método planejado teve um desempenho superior a todos os métodos. Segundo os autores, a programação genética foi a principal responsável por esses resultados, pois o algoritmo inicialmente utilizava amostras aleatórias para preencher as lacunas antes de ser submetido a processos genéticos. Os resultados confirmaram que estratégias baseadas em algoritmos evolutivos são alternativas viáveis para o tratamento

de valores ausentes.

Shahzad, Rehman e Ahmed (2017), em seu estudo, "Imputação de Dados Ausentes usando Algoritmo Genético para Aprendizado Supervisionado", empregaram algoritmos genéticos para buscar valores plausíveis na imputação de dados ausentes. Uma estratégia interessante adotada neste estudo é o uso do ganho de informação para observar como as soluções são encontradas à medida que o processo avança. Em um experimento com cinco conjuntos de dados que originalmente continham valores ausentes, o método proposto foi comparado com outras abordagens de imputação: média, menor valor, maior valor, zero e IM. Eles utilizaram as seguintes medidas de desempenho: acurácia preditiva, precisão, *recall*, *F-measure* e a área sob a curva da Característica Operacional do Receptor - ou *Receiver Operating Characteristics* (ROC), com os seguintes classificadores: *NB-tree*, PART, JRIP, *Naïve Bayes*, KNN e J48. Os autores observaram que o método baseado em AG apresentou resultados promissores e funcionou bem em conjuntos de dados com uma alta porcentagem de valores ausentes.

Em Lobato et al. (2015a), um algoritmo chamado MOGAImp foi proposto para conjuntos de dados de imputação múltipla com base em algoritmos genéticos. Uma das estratégias interessantes deste trabalho é aplicar uma abordagem multiobjetivo, que até então não havia sido adotada na literatura para a análise de desempenho de técnicas de imputação. Essa abordagem envolve o emprego simultâneo de duas ou mais medidas de avaliação. A abordagem pode ser explicada pelo fato de que existem distinções entre várias medidas de desempenho porque, enquanto uma aumenta, a outra diminui. No caso do MOGAImp, foram utilizadas duas medidas conflitantes: a acurácia do classificador e a acurácia preditiva do método de imputação, calculadas usando *Normalized Root-Mean-Square Error* (NRMSE) e a fronteira de Pareto.

Outro fator crítico no estudo conduzido por Lobato et al. (2015a) diz respeito à inicialização da população, que utiliza um conjunto de soluções candidatas com base em cada atributo. O conjunto de soluções envolve agrupar todos os possíveis valores do conjunto de dados para o atributo que possui um valor ausente (comparando lexicograficamente duas strings em casos de variáveis categóricas). O método foi comparado experimentalmente com outras técnicas bem conhecidas na literatura, empregando benchmarking por meio de vários bancos de dados com valores ausentes. Os resultados demonstraram que o método alcançou desempenho competitivo e, segundo os autores, mostrou potencial para aplicações do mundo real. No entanto, é necessária uma alta potência computacional para lidar com os valores ausentes individualmente com o MOGAImp e por meio do conjunto de soluções. Além disso, essa estratégia é uma excelente alternativa para uma mistura de materiais genéticos. Portanto, ela foi adotada no EvoImp como uma referência para operações de mutação.

Em Lobato (2016), os autores criaram um esquema baseado em algoritmos genéticos,

que serviu como referência para o desenvolvimento e análise do método empregado neste estudo. A estratégia, nomeada como MultImp, prevê múltiplas imputações de conjuntos de dados em um modelo de classificação multirrótulo. Neste estudo, os autores conduziram experimentos utilizando quatro bancos de dados que foram inicialmente preenchidos. Posteriormente, 5% dos valores ausentes foram adicionados por meio do mecanismo MCAR. O *Binary Relevance* foi empregado como classificador multirrótulo, com C4.5 como parâmetro. No cenário de teste, o MultImp foi comparado com dois outros métodos de imputação (*K-Nearest Neighbors Imputation* e *Most Common*) e avaliado lexicograficamente usando as seguintes medidas: *Exact Match*, Acurácia, e *Hamming Loss*. Os resultados preliminares deste estudo mostraram-se promissores, especialmente no caso de EM, onde o desempenho alcançado pelo método foi melhor em todos os conjuntos de dados utilizados, justificando a adoção da abordagem lexicográfica.

A Tabela 3 apresenta um resumo dos trabalhos discutidos nesta Seção, os quais utilizam AG no tratamento de dados ausentes. Essa tabela é formada pelo nome dos métodos (traduzidos para o Português) e destacando os pontos fortes e limitações identificadas. Buscou-se superar cada uma dessas limitações no método proposto nesta tese, as quais são sumarizadas abaixo:

- Quanto à complexidade computacional, o EvoImp possui uma complexidade menor em comparação com os trabalhos discutidos, e esta é detalhada na Seção 5.2.2;
- No que se refere à inicialização da população do AG, é importante destacar que essa é uma das melhorias desenvolvidas frente ao método base (LOBATO, 2016). Essa abordagem é explicada na Seção 4.2;
- Quanto à configuração experimental, foram realizados testes robustos com uma quantidade considerável de bases de dados e métodos de imputação. Essas escolhas foram fundamentadas na literatura, e uma discussão mais aprofundada sobre esses pontos será apresentada no Capítulo 5;
- Com relação aos estudos que se concentram, principalmente, em ML em vez de valores ausentes, até onde sabemos, não há trabalho abordando valores ausentes no espaço de recursos preditivos em um cenário de *Multi-label Classification*. Portanto, isso constitui uma das contribuições do presente estudo.

Tabela 3 – Sumarização dos trabalhos correlatos destacando os métodos, pontos fortes e limitações de cada trabalho.

Referência	Método	Pontos Fortes	Limitações
(TRAN; ZHANG; ANDREAE, 2015)	GPMI - Programação Genética para Imputação de Dados Ausentes	- Aplicou uma estratégia de imputação múltipla; - Realizou estimação de valores ausentes utilizando técnicas de regressão.	- Alta complexidade computacional; - A população inicial é inicializada aleatoriamente, aumentando o espaço de busca.
(SHAHZAD; REHMAN; AHMED, 2017)	Algoritmo Genético para Aprendizado Supervisionado	- Utilizou Ganho de Informação; - Comparou com abordagens de imputação múltipla;	- A configuração experimental considerou apenas conjuntos de dados pequenos; - A comparação foi feita apenas contra métodos simples de imputação.
(LOBATO et al., 2015a)	MOGAImp - Algoritmo Genético Multi-Objetivo para Imputação de Dados Ausentes	- Adotou uma estratégia multi-objetivo, lidando com medidas de avaliação conflitantes; - Adequado para aplicações do mundo real; - Lida com conjuntos de dados de atributos mistos (categóricos ou contínuos).	A população inicial é inicializada aleatoriamente, aumentando o espaço de busca.
(LOBATO, 2016)	MultImp - Imputações Múltiplas baseadas em algoritmos genéticos	- Prevê múltiplas imputações em classificação multirrótulo; - Apresenta desempenho competitivo.	- A configuração experimental considerou apenas conjuntos de dados pequenos; - A comparação foi feita apenas contra KNNI e métodos de imputação MC.

4 EVOIMP - MÉTODO PROPOSTO

Uma vez que o EvoImp é baseado em um algoritmo genético, as seguintes descrições explicam como o EvoImp foi mapeado e configurado dentro da estrutura de AG: a) a codificação dos indivíduos, b) a formação da população inicial, c) a configuração dos operadores genéticos e d) a definição da função de aptidão. A Figura 5 apresenta um exemplo prático dessa estrutura, que será detalhado nas subseções seguintes.

4.1 CODIFICAÇÃO DOS INDIVÍDUOS E POPULAÇÃO INICIAL

A codificação de indivíduos do EvoImp ocorreu da seguinte forma: as variáveis nos conjuntos de dados representam genes individuais. Genes inicialmente marcados com "?" representam os valores ausentes (Figura 5 (a)). Cada indivíduo é representado por uma instância completa ("realizada") dos bancos de dados (Figura 5 (b)). O fenótipo consiste em valores imputados, enquanto o genótipo representa esses valores em forma binária, como ilustrado na Figura 5 (c).

A população inicial compreendia cinco métodos simples de imputação para a geração de cada indivíduo (Figura 5 (d))[1]. Todos os métodos de imputação são bem conhecidos e estabelecidos na literatura (LUENGO; GARCÍA; HERRERA, 2012): KMI, WKNNI, CMC e MC. Os parâmetros para os métodos KNNI, WKNNI e KMI seguiram as diretrizes estabelecidas pelos autores. Esse tipo de inicialização da população foi adotado no EvoImp para reduzir o espaço de busca e, consequentemente, os custos computacionais.

Ao contrário do MOGAImp (LOBATO et al., 2015a), que emprega inicialização aleatória da população inicial, o método proposto otimiza imputações simples por meio de processos evolutivos para realizar múltiplas imputações. Essa abordagem reduz o espaço de busca e introduz um método inovador. Essa redução no espaço de busca é particularmente benéfica em cenários onde o custo computacional é crítico nos cálculos da função objetivo, como na classificação multirrótulo.

É também digno de nota que o trabalho apresentado possui dois conteúdos inovadores: 1) o uso de métodos simples de imputação como solução *a priori*, reduzindo o espaço de busca; 2) o tratamento de valores ausentes no cenário multirrótulo. Até onde se sabe, não há estudo semelhante na literatura.

[1] Para melhor ilustrar a troca genética dos *datasets* completos com dados imputados, optou-se por representar cada *dataset* utilizando uma cor diferente.

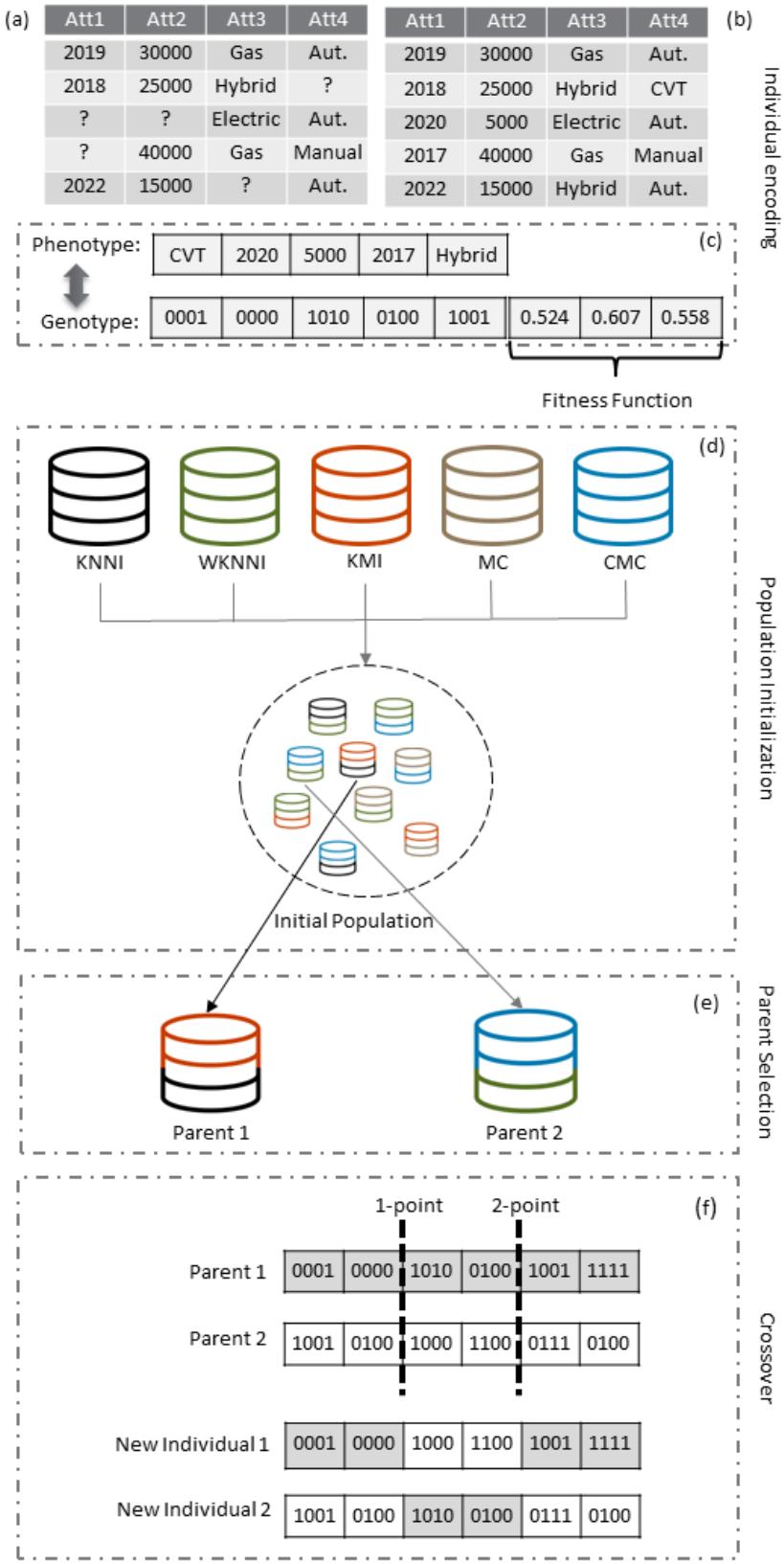


Figura 5 – Estrutura do AG do EvoImp. (a) Conjunto de dados com valores ausentes; (b) Um conjunto de dados completo com dados imputados. (c) Fenótipo com os valores correspondentes ao espaço de dados ausentes; Genótipo: representação dos genes em código binário e os valores das medições usadas na função de aptidão. (d) Ilustração da inicialização da população. (e) Seleção para o cruzamento. (f) Aplicação do cruzamento aos dois indivíduos selecionados.

4.2 OPERADORES GENÉTICOS

A seleção de indivíduos envolve um torneio no qual dois (ou mais) membros da população anterior são selecionados, e o melhor é escolhido com base no valor de aptidão, como ilustrado na Figura 5 (e). Esse procedimento foi seguido até que um número limitado de indivíduos da geração atual fosse obtido. O melhor indivíduo é sempre selecionado por meio do elitismo (MIRJALILI, 2019).

Na literatura, numerosos métodos para ajuste e controle de parâmetros foram propostos e analisados. Karafotias, Hoogendoorn e Eiben (2015) descrevem alguns desses métodos e discute várias tendências e desafios no campo. Especificamente, Reynoso-Meza et al. (2011) conduziram experimentos para encontrar configurações apropriadas para esses parâmetros ao aplicar algoritmos evolutivos a uma classe de problema multiobjetivo. Eles concluíram que determinar o valor do fator de escala pode ser difícil e é altamente dependente do problema específico.

Considerando essas descobertas, testes iniciais foram conduzidos para definir os parâmetros usados em nosso estudo. Em linha com o trabalho de Reynoso-Meza et al. (2011), a probabilidade inicial de *Crossover* foi delimitada para [0.8, 1.0], seguindo a proposta padrão para problemas não separáveis como o abordado nesta pesquisa. O EvoImp emprega uma probabilidade de *crossover* para 80% dos indivíduos usando um operador de *n-point crossover* (SEMENKIN; SEMENKINA, 2012), como mostrado na Figura 5 (f). Isso também está em consonância com o trabalho de Lobato et al. (2015a).

O processo de mutação é realizado em 20%[2] dos indivíduos escolhidos aleatoriamente, exceto pelo melhor. Para cada indivíduo a ser mutado, o valor imputado é trocado por um valor candidato. A mutação é aplicada apenas aos genes que contêm valores ausentes. Para realizar isso, cada atributo no conjunto de dados tem um conjunto de soluções, conforme mostrado na Tabela 4. Esse conjunto é formado considerando todas as opções de resposta possíveis para aquele atributo no conjunto de dados avaliado.

A Tabela 4a exibe um conjunto de dados fictício contendo cinco registros e quatro atributos: “Ano”, “Gênero”, “Idade” e “Tem Crédito”. Alguns valores no conjunto de dados estão ausentes e são representados por “?”. A Tabela 4b lista os valores possíveis para cada atributo. Por exemplo, o atributo “Ano” pode ter valores 1998, 2005 ou 2010; e o atributo “Gênero” pode ter valores M ou F. O mesmo raciocínio é aplicado aos outros atributos.

Lobato et al. (2015a) adotaram essa técnica para iniciar a primeira população do MOGAImp. O operador de mutação não foi implementado no MultImp. A falta dele causou uma convergência prematura, limitando a robustez do método. Esse operador é

[2] A taxa de mutação é mais elevada do que as taxas normalmente utilizadas. Nesse caso, considerando que a população inicial é obtida por outros métodos, experimentos de parametrização demonstraram que uma taxa de mutação mais alta proporciona melhores resultados, garantindo uma rápida convergência. O resultado dos testes realizados para um dataset de calibração é apresentado no Apêndice A.

Tabela 4 – Soluções candidatas para cada atributo usado no processo de mutação.

Ano	**Gênero**	**Idade**	**Tem Crédito**
2010	M	25	?
?	F	?	Sim
2005	?	32	?
1998	M	?	Sim
?	?	30	Não

(a) Exemplo de conjunto de dados com valores ausentes.

Atributo	**Valores**
Ano	1998, 2005, 2010
Gênero	M, F
Idade	25, 30, 32
Tem Crédito	Sim, Não

(b) Conjunto de valores possíveis

uma das principais diferenças entre o MultImp e o EvoImp. Em outras palavras, o método proposto implementa uma estratégia para evitar mínimos locais.

O processo de busca e otimização do algoritmo ocorre ao longo de gerações predeterminadas. A população entra em uma fase de crescimento, começando com o número de métodos de MI adotados na inicialização da população e aumentando por meio de seu cruzamento. Essa estratégia visa fornecer diversidade à população. Na segunda fase, a população é gradualmente reduzida, alcançando o mesmo tamanho inicial da população, permitindo a análise para escolher qualitativamente a melhor solução.

4.3 FUNÇÃO DE APTIDÃO

Conforme mencionado anteriormente, o método foi avaliado em um cenário de MLC. Para isso, o EvoImp realiza um processo de classificação para cada indivíduo. O objetivo é analisar o desempenho do classificador e, consequentemente, a eficiência da imputação de dados.

Três medidas de desempenho foram adotadas para avaliar o classificador, assim como o MultImp: *Exact Match*, Acurácia e *Hamming Loss*. Essas medidas foram usadas em ordem lexicográfica; em outras palavras, essa abordagem prioriza todos os objetivos do problema e, em seguida, tenta satisfazê-los, mantendo uma lista de prioridades (GONZÁLEZ et al., 2021). Assim, a aptidão (f) para a solução do problema pode ser expressa como Eq. 4.1:

$$f = [f_0, f_1, ..., f_{n-1}] \in \mathbb{R}^n \tag{4.1}$$

na qual n é o número de objetivos definidos; f_n é um objetivo de otimização. Dadas duas avaliações de aptidão f_1 e f_2 e um limite de precisão t, a relação lexicográfica entre elas (notada como $\prec_l$ e $\preceq_l$) pode ser definida (GONZÁLEZ et al., 2019):

$$\begin{aligned} f_1 \prec_l f_2 \Leftrightarrow \exists k \in [0, n_0) \cap \mathbb{N} : f_1^k < f_2^k \wedge \left| f_1^k - f_2^k \right| \geqslant t \\ \wedge \left| f_1^i - f_2^i \right| < t \ \forall i < k \end{aligned} \tag{4.2}$$

$$f_1 =_l f_2 \Leftrightarrow \left|f_1^i - f_2^i\right| < t \ \forall i \in [0, n_o) \cap \mathbb{N} \tag{4.3}$$

$$f_1 \preceq_l f_2 \ \Leftrightarrow \ f_1 \prec_l f_2 \ \ \vee f_1 =_l f_2 \tag{4.4}$$

Como pode ser observado, a Eq. 4.2 mostra que $f_1 \prec_l f_2$, o que significa que f_1 é lexicograficamente menor que f_2. Essa relação é estabelecida quando existe um índice k no intervalo $[0, n_0) \cap \mathbb{N}$, tal que $f_1^k < f_2^k$, indicando que o componente k-ésimo de f_1 é menor que o componente k-ésimo de f_2. Além disso, a diferença entre f_1^k e f_2^k é maior ou igual a t. Isso garante que os componentes k-ésimos diferem significativamente pelo menos t. Finalmente, as diferenças absolutas entre os componentes correspondentes f_1^i e f_2^i devem ser menores que t para todos os i menores que k. Em essência, essa relação significa que f_1 é superior a f_2 em termos de alguns objetivos.

A Eq. 4.3 determina a igualdade na ordem lexicográfica ($f_1 =_l f_2$). Isso ocorre quando as diferenças absolutas entre os componentes correspondentes f_1^i e f_2^i são todas menores que t para todos os i no intervalo $[0, n_0) \cap \mathbb{N}$. Em outras palavras, f_1 e f_2 são considerados iguais em relação ao seu desempenho em relação aos objetivos.

Finalmente, a Eq. 4.4 apresenta $f_1 \preceq_l f_2$, o que significa que f_1 é menor ou igual a f_2 na ordem lexicográfica. Ela combina as relações $\prec_l$ e $\preceq_l$, indicando que f_1 é tanto melhor quanto igual a f_2 em termos dos objetivos definidos.

Essa abordagem permite adicionar diferentes medidas de desempenho a uma única avaliação (GONÇALVES; PLASTINO; FREITAS, 2013). É semelhante à abordagem lexicográfica clássica, mas, uma vez que algoritmos evolutivos são adotados, ótimos locais podem ser evitados (GONZÁLEZ et al., 2019).

4.4 O ALGORITMO EVOIMP

O algoritmo do EvoImp é apresentado no Algoritmo 2. Outro formato de apresentação pode ser observado no fluxograma (Figura 7) presente no Apêndice B. O EvoImp inicia a execução criando e avaliando indivíduos para a população inicial. Os conjuntos de dados são inicialmente imputados usando métodos simples de imputação: KNNI, CMC, MC, KMI e WKNNI (linhas 1-5). Em seguida, o algoritmo inicia o processo de aplicação dos operadores genéticos enquanto o critério de parada não for atingido (por exemplo, o número de gerações) (linhas 6-19).

No início de cada iteração, o algoritmo realiza a avaliação dos indíviduos (linha 7) e realiza a ordenação da população utilizando a ordem lexicográfica (linha 8), conforme descrita na subseção 4.3.

Antes de começar a gerar a nova população, o indivíduo elitista é sempre passado para a próxima geração (linha 9). A seleção é realizada usando o operador de seleção de torneio (linha 11). Dois indivíduos são sorteados aleatoriamente nesse processo. Esses dois

Algoritmo 2: EvoImp

Input: conjuntos de dados com VAs e parâmetros (ver Tabela 6)
Output: conjuntos de dados completos

```
1  foreach Método de Imputação Simples do
2      Gerar um novo Indivíduo: individuo;
3      Avaliar o individuo;
4      Adicionar individuo à População Atual: PopAtual ← individuo;
5  end
6  while Critério de parada não alcançado do
7      Avaliar a PopAtual;
8      Ordenar PopAtual usando a ordem lexicográfica;
9      Adicionar à População Atual o Melhor Indivíduo: PopAtual ← MelhorIndividuo;
10     while PopAtual < Número de indivíduos da nova geração do
11         Selecionar Pais;
12         Aplicar Crossover;
13         Adicionar o Indivíduo à População Atual: PopAtual ← individuo;
14     end
15     while Número de indivíduos mutados < 20% dos indivíduos da nova geração do
16         Escolher aleatoriamente um indivíduo da População Atual;
17         Aplicar Mutação;
18     end
19 end
20 return MelhorIndividuo;
```

pais trocam material genético usando um operador de cruzamento (linha 12). Essas etapas são repetidas até que a população esteja completa (critério delimitado na linha 10). Em seguida, a mutação segue a taxa estabelecida (linhas 15-18). O processo iterativo continua até que o critério de parada seja atingido. Por fim, o algoritmo retorna o indivíduo que alcança o melhor desempenho (linha 20).

Em resumo, o EvoImp adota a configuração para a parametrização do MultImp (LOBATO, 2016), exceto pelo operador de mutação, conforme mencionado anteriormente. Além disso, foram corrigidos bugs e otimizado o código, levando em consideração a manutenibilidade e a reutilização. Além disso, foi implementado a estratégia lexicográfica e expandiu-se os testes computacionais, ampliando a contribuição técnico-científica do presente trabalho.

5 EXPERIMENTOS COMPUTACIONAIS

Neste capítulo são apresentados os detalhes relacionados aos *datasets* escolhidos, bem como a configuração dos experimentos realizados. Por fim, é especificada a complexidade computacional do método.

5.1 DATASETS

Os experimentos foram projetados usando seis conjuntos de dados multirrótulos do repositório de aprendizado de máquina da UCI[1]. A quantidade de conjuntos de dados está em conformidade com a revisão da literatura realizada por Chiu et al. (2022), que mapearam 48 artigos relacionados a experimentos no contexto de imputação de dados.

O trabalho de Chiu et al. (2022) mostra que a maioria dos artigos (77%) utiliza até seis conjuntos de dados (*datasets*) em experimentos. Uma descoberta interessante de Chiu et al. (2022) é que o Repositório de Aprendizado de Máquina da UCI é o mais utilizado. Em relação às características dos conjuntos de dados, a maioria utiliza conjuntos de dados em pequena escala, que contêm menos de 15 atributos e 800 instâncias. A Tabela 5 mostra os conjuntos de dados utilizados e suas características.

Tabela 5 – Datasets utilizados nos experimentos.

Dataset	Cod.	Domínio	Inst.	Nominal Atr.	Númerica Atr.	Total Atr.	Rótulos	Cardinal.	Densidade
Birds	b	Áudio	645	2	258	260	19	1,014	0,053
Cal500	c	Música	502	0	68	68	174	26,044	0,150
Emotions	e	Música	593	0	72	72	6	1,869	0,311
Flags	f	Imagem	194	9	10	19	7	3,392	0,485
Scene	s	Imagem	2407	0	294	294	6	1,074	0,179
Yeast	y	Biologia	2417	0	103	103	14	4,237	0,303

Em relação aos conjuntos de dados multirrótulos, deve-se mencionar os trabalhos de Esmaeili et al. (2020) e Wang, Lin e Liu (2019). Esses estudos, assim como o EvoImp, utilizaram conjuntos de dados obtidos no repositório da UCI e formatados usando a biblioteca Mulan[2]. Os conjuntos de dados utilizados nesses artigos possuem características semelhantes (cardinalidade, densidade e número de instâncias) aos escolhidos. Essa

[1] https://archive.ics.uci.edu/
[2] http://mulan.sourceforge.net/

observação destaca a consonância na configuração experimental com o estado da arte e a potencial aplicabilidade do EvoImp em problemas do mundo real.

5.2 CONFIGURAÇÃO EXPERIMENTAL

Nos experimentos, os valores ausentes foram adicionados artificialmente a cada conjunto de dados com as seguintes taxas: 5%, 10%, 15%, 20%, 25% e 30%. Esse processo de "amputação" foi realizado usando o mecanismo MCAR, conforme descrito em Santos et al. (2019). A configuração experimental completa consistiu em 36 conjuntos de dados com dados ausentes, e esses conjuntos de dados foram submetidos a uma avaliação comparativa. Essa avaliação envolveu cinco métodos simples de imputação: KNNI, CMC, MC, KMI e WKNNI.

Os seguintes métodos de classificação foram utilizados para as tarefas de aprendizado multirrótulo[3]: BR, *Hierarchy of Multi-label classifiER* (HOMER), *Multi-Label K-Nearest Neighbors* (ML-KNN), *Classifier Chains* (CC) e *Ensembles of Classifier Chains* (ECC) (READ et al., 2011; TSOUMAKAS; KATAKIS; VLAHAVAS, 2008). A validação cruzada *K-fold* foi utilizada para a avaliação do modelo de classificação (aprendizado e teste). A Tabela 6 resume os parâmetros gerais que foram utilizados nos experimentos.

Tabela 6 – Configurações dos parâmetros utilizadas nos experimentos.

Parâmetro	Valor
População inicial	cinco indivíduos (conjuntos de dados imputados)
Gerações	7
Taxa de crossover	80% dos indivíduos
Taxa de mutação	20%
Tipo de seleção	Torneio (tamanho=2)
Métodos de imputação	KNNI, CMC, MC, KMI e WKNNI
Taxas de valores ausentes	5, 10, 15, 20, 25 e 30%
Método de ocorrência de MV	MCAR
Algoritmos de MLC	BR, HOMER, ML-KNN, CC e ECC
Validação cruzada *K-fold*	k=10

5.2.1 Implementação

Foram programadas duas versões do EvoImp, a versão utilizada nos experimentos foi programado na linguagem Java, versão 8.1, com base nos trabalhos de Lobato et al. (2015b), Lobato (2016). Uma versão na linguagem *Python* também foi programada para fins de validação. A seguir, os componentes de terceiros da versão em Java são detalhados:

[3] Como forma de comparação, realizamos os testes do EvoImp e dos classificadores com os datasets sem dados ausentes (*baseline*). Os resultados desse teste são encontrados no Apêndice C.

- Para os classificadores multirrótulos, foi utilizada a biblioteca Mulan[4] (TSOUMAKAS et al., 2011). Esta biblioteca também contém alguns classificadores implementados no Weka[5] (FRANK; HALL; WITTEN, 2016).
- Os métodos de imputação múltipla de dados usados para formar a população inicial do EvoImp e nas análises comparativas são implementados no software KEEL[6] (TRIGUERO et al., 2017).

É importante destacar que o AG usado no EvoImp foi totalmente implementado, apesar do KEEL fornecer um framework para computação evolutiva. Essa decisão de *design* teve como objetivo proporcionar mais controle sobre os experimentos.

Conforme mencionado, a versão em *Python* foi implementada para se validar os resultados. Para a imputação de dados e classificação multirrótulo a biblioteca *scikit-learn* foi escolhida por ter todos os métodos utilizados nos experimentos implementados. Em um trabalho futuro pretende-se disponibilizar a ferramenta em *Python* considerando o crescente uso dessa linguagem de programação em projetos envolvendo ciência de dados.

5.2.2 Complexidade Computacional do Método

A complexidade computacional é outro aspecto crucial a ser considerado na implementação deste método proposto. Ela desempenha um papel vital na determinação da viabilidade e eficiência da aplicação de técnicas bioinspiradas para resolver problemas de otimização. Abordar essa preocupação e reduzir a complexidade computacional aprimora a aplicabilidade e escalabilidade do algoritmo. Como resultado, torna-o mais adequado para lidar com conjuntos de dados maiores e cenários de otimização complexas, especialmente em tarefas de classificação multirrótulo. Nesse contexto, a complexidade computacional do EvoImp ($O(Method)$) está estruturada em três processos principais:

- Geração da população inicial usando Métodos Simples de Imputação ($O(SIM_i)$);
- Operações genéticas: seleção, crossover e mutação ($O(GenOp)$);
- Classificação para medir o desempenho dos indivíduos ($O(Class_j)$).

Portanto, a complexidade do EvoImp pode ser descrita pela Eq. 5.1:

$$O(Method) = \sum_{i=1}^{n} O(SIM_i) + O(GenOp) + \sum_{j=1}^{m} O(Class_j) \tag{5.1}$$

[4] https://mulan.sourceforge.net/
[5] https://www.cs.waikato.ac.nz/ml/weka/index.html
[6] http://www.keel.es/

onde $i \in (1, \ldots, n)$ denota o número de métodos de imputação adotados, e $j \in (1, \ldots, m)$ denota o número de métodos de classificação usados para as tarefas de aprendizado multirrótulo.

As complexidades $O(SIM_i)$ e $O(GenOp)$ para o problema estudado tiveram um baixo impacto na equação. Isso ocorre porque os atributos que influenciam essas complexidades têm um valor baixo (por exemplo, o número de indivíduos, o número de gerações, o número de valores ausentes, e outros). Neste sentido, conforme explica Cormen et al. (2022), essas funções são dominadas assintoticamente por outras. Nesse contexto, somente as funções que aumentam o tempo de execução à medida que o tamanho da entrada aumenta sem limite devem ser levadas em consideração. Nesse caso, as complexidades $O(SIM_i)$ e $O(GenOp)$ são muito menores que $O(Class_j)$. Esse fato foi confirmado durante os experimentos computacionais. Nesse sentido, a equação atualizada desprezando as complexidades de $O(SIM_i)$ e $O(GenOp)$ fica como (Eq. 5.2):

$$O(Method) = \sum_{j=1}^{m} O(Class_j) \tag{5.2}$$

Por outro lado, segundo Bogatinovski et al. (2022), a complexidade da maioria dos classificadores multirrótulo depende do tamanho do banco de dados (X) e, especialmente, do número de rótulos (Y). Levando, também, em consideração que ao se somar complexidades de ordem de grandeza diferentes, os termos de ordem menor devem ser eliminados e somente a maior ordem deverá representar a complexidade (CORMEN et al., 2022). Nesse contexto, pode-se concluir que a complexidade do método é (Eq. 5.3):

$$O(Method) = MAX(O(Class_j)) \tag{5.3}$$

Entre os classificadores escolhidos para este trabalho, HOMER apresentou o pior desempenho, o que pode ser justificado pelo processo de agrupamento balanceado (BANERJEE; GHOSH, 2006).

6 RESULTADOS E ANÁLISES

Neste capítulo são explicitados os resultados dos experimentos realizados e, também, uma discussão destacando alguns pontos de performance obtidos.

6.1 RESULTADOS

Esta Seção examina os resultados obtidos a partir dos experimentos computacionais. Os dados exibidos nas tabelas seguintes mostram as diferenças de desempenho entre os métodos para cada porcentagem de valores ausentes analisados (5%, 10%, 15%, 20%, 25% e 30%). Os melhores resultados estão destacados em negrito para facilitar a visualização. As métricas (*Exact Match* (↑), *Accuracy* (↑) e *Hamming Loss* (↓)) são apresentadas com esses símbolos, onde (↑) indica que valores mais altos refletem melhor desempenho, e (↓) indica que valores mais baixos representam melhor desempenho.

Binary Relevance

Na aprendizagem realizada com o classificador BR, os resultados mostraram que o EvoImp foi numericamente superior (Tabela 7). Na avaliação EM, o EvoImp superou seus concorrentes em 35 dos 36 conjuntos de dados avaliados (97,22%). O método proposto demonstrou desempenho superior em relação aos outros em 18 cenários de conjuntos de dados (50%) na medida de avaliação *Accuracy*. Finalmente, considerando o HL, o EvoImp superou os métodos de referência em 16 conjuntos de dados (44,44%).

É fundamental destacar as prioridades adotadas na ordem lexicográfica do EvoImp, priorizando a avaliação com EM, conforme mencionado na Subseção "Função de Aptidão" 4.3, o que explica a diminuição de desempenho nas métricas ACC e HL considerando o classificador *Binary Relevance*.

Hierarchy of Multi-label classifiER

Os resultados para o classificador HOMER são apresentados na Tabela 8. Ao analisar os resultados, é possível observar que o EvoImp também se destaca em 35 dos 36 conjuntos de dados utilizados nos experimentos (97,22%) em relação à métrica EM. Esses resultados corroboram com os obtidos a partir do classificador *Binary Relevance*.

Tabela 7 – Resultados para o classificador BR.

		Exact Match (↑)						Accuracy (↑)						Hamming Loss (↓)					
%[1]	Db[2]	KMI[3]	KNNI[3]	MC[3]	CMC[3]	WKNNI[3]	EvoImp	KMI	KNNI	MC	CMC	WKNNI	EvoImp	KMI	KNNI	MC	CMC	WKNNI	EvoImp
5	b	46.67	48.54	51.32	51.22	51.31	**52.57**	58.01	61.82	62.33	60.91	62.32	**62.94**	05.52	05.21	05.05	05.06	05.13	**05.01**
	c	33.77	34.92	33.93	34.13	34.86	**35.24**	44.04	44.88	44.87	**45.12**	44.75	44.73	15.81	15.92	15.55	**15.42**	15.91	15.87
	e	50.78	50.82	48.83	51.21	50.71	**51.44**	54.04	56.74	53.42	56.77	56.54	**57.24**	24.92	25.02	25.94	24.78	25.32	**24.51**
	f	58.82	58.88	60.22	59.41	58.44	**61.94**	**70.02**	68.75	69.82	69.51	68.63	68.48	**27.42**	28.49	27.86	28.03	28.63	27.76
	s	53.59	56.92	51.04	52.81	56.93	**58.12**	51.23	55.85	48.84	50.71	55.27	**56.68**	14.14	13.34	14.67	13.93	13.31	**13.02**
	y	50.67	49.72	50.17	50.53	50.64	**51.32**	59.90	59.92	60.33	61.17	59.94	**61.78**	24.83	24.96	24.22	23.92	24.89	**23.84**
10	b	46.81	48.43	47.71	46.82	46.33	**49.87**	56.51	60.02	**60.24**	56.93	57.42	60.11	05.74	05.32	**05.12**	05.44	05.51	05.32
	c	35.54	36.14	34.71	34.63	36.46	**36.63**	44.45	45.62	45.36	45.22	45.78	**45.85**	15.52	15.53	**14.92**	15.04	15.52	15.51
	e	47.54	49.87	44.82	49.63	48.25	**50.97**	53.62	53.24	47.04	52.27	**54.43**	54.12	26.72	26.05	28.13	**25.22**	26.56	25.96
	f	57.52	60.22	57.43	58.41	60.34	**61.64**	70.43	73.12	70.99	68.73	73.34	**73.72**	26.93	26.13	26.22	28.31	25.96	**25.22**
	s	52.65	56.81	47.68	51.82	57.03	**58.34**	51.12	56.13	43.61	48.16	56.56	**57.37**	13.94	12.94	15.25	13.91	12.85	**12.34**
	y	49.56	50.64	48.81	48.42	50.22	**51.93**	59.23	60.31	59.66	59.13	60.37	**61.17**	24.85	24.59	**24.21**	24.42	24.67	24.43
15	b	44.42	45.87	47.32	43.86	44.62	**47.36**	56.31	57.38	59.03	54.58	54.52	**59.14**	05.63	05.26	04.98	05.62	05.31	**04.92**
	c	34.73	35.41	36.06	36.01	35.95	**36.17**	43.91	43.97	47.03	**47.55**	44.72	47.36	14.92	15.43	14.19	**14.04**	15.32	14.08
	e	48.41	49.43	44.09	47.61	48.13	**50.32**	52.73	55.32	46.63	53.51	55.74	**56.22**	26.31	25.59	27.05	25.13	**24.92**	25.46
	f	61.74	61.76	59.03	61.85	63.71	**64.35**	73.11	73.45	72.14	72.16	74.95	**75.07**	25.91	24.46	25.95	24.98	23.06	**22.87**
	s	50.72	58.16	46.35	49.64	58.19	**58.95**	48.94	**57.95**	41.51	47.15	57.14	57.65	14.24	12.56	15.41	13.54	12.51	**12.21**
	y	47.50	50.94	48.71	49.95	50.94	**51.44**	57.61	60.25	60.27	60.76	59.91	**61.09**	24.41	24.34	23.36	**23.31**	24.74	24.38
20	b	43.04	45.84	43.28	42.51	43.79	**47.37**	54.66	**58.27**	53.94	52.26	55.51	57.65	05.51	**04.86**	05.15	05.52	05.20	04.97
	c	35.82	35.41	35.55	35.47	35.45	**36.45**	44.02	43.84	**47.36**	46.38	43.43	44.31	14.44	15.17	**13.51**	13.64	15.27	14.94
	e	**50.12**	45.44	40.82	48.46	46.23	48.92	**52.38**	50.34	39.94	51.17	50.49	51.24	25.36	27.28	27.51	24.04	27.42	**23.85**
	f	57.93	61.05	59.61	62.51	60.25	**63.37**	70.02	71.76	71.74	71.73	**72.87**	72.12	27.14	24.46	26.04	24.97	**23.92**	24.56
	s	45.52	57.74	43.61	47.48	58.45	**58.74**	42.01	57.26	39.08	44.34	57.63	**57.73**	15.23	12.36	15.48	13.69	**11.85**	11.93
	y	49.25	50.92	50.16	49.23	51.57	**51.78**	58.53	61.06	61.67	**61.75**	60.85	60.82	24.52	24.07	**22.01**	22.36	23.92	23.97
25	b	43.45	43.44	44.15	42.14	43.94	**44.75**	55.47	56.25	56.56	52.57	**57.78**	56.91	05.15	05.01	04.84	05.01	04.83	**04.73**
	c	37.76	37.21	37.75	36.26	36.47	**37.85**	45.81	44.85	50.56	47.77	44.21	**50.56**	13.94	14.98	12.64	13.01	15.06	**12.62**
	e	43.23	45.14	38.22	46.57	45.06	**47.72**	44.35	**50.13**	37.21	49.32	49.41	50.12	25.94	26.82	26.31	**23.48**	27.09	25.24
	f	60.25	58.92	62.23	60.24	59.71	**63.46**	72.51	72.65	73.13	70.32	72.47	**73.04**	27.04	25.83	26.08	27.42	**25.15**	25.31
	s	47.14	59.36	39.94	45.16	58.87	**60.16**	43.34	58.68	36.53	42.32	58.74	**59.14**	14.61	11.73	16.02	13.61	12.09	**11.72**
	y	49.72	51.75	48.51	49.21	51.21	**51.96**	**62.96**	61.65	61.74	61.71	61.08	61.74	**21.42**	23.21	22.05	21.84	23.68	23.29
30	b	39.15	39.41	42.27	42.31	41.03	**43.36**	50.25	51.71	52.82	52.76	**53.48**	53.03	05.51	05.48	**04.94**	05.22	05.21	05.14
	c	37.13	35.92	37.82	37.65	35.43	**38.01**	45.21	43.24	**49.92**	49.01	43.26	47.57	13.62	14.73	**12.02**	12.21	14.62	12.97
	e	44.52	45.96	41.12	48.91	48.03	**49.62**	48.63	49.22	38.85	53.67	53.65	**54.37**	26.52	26.81	26.24	24.06	26.13	**23.81**
	f	62.04	62.42	59.71	64.74	63.11	**65.54**	74.32	74.23	74.74	75.13	74.58	**75.46**	24.68	24.42	24.84	23.72	24.73	**23.26**
	s	44.53	59.32	39.46	44.18	58.97	**59.62**	40.94	59.01	35.78	41.92	**59.38**	59.23	15.37	11.79	15.32	13.52	11.74	**11.62**
	y	50.13	51.82	49.71	49.04	51.57	**52.71**	59.96	62.23	62.72	**63.02**	62.04	62.84	23.52	22.83	**20.92**	21.04	22.94	22.52
Avg rank	-	5	2	6	4	3	**1**	6	3	5	4	2	**1**	6	5	3	2	4	**1**

[1] "%" refere-se à porcentagem de dados ausentes analisada (5%, 10%, 15%, 20%, 25% e 30%).

[2] "Db" refere-se aos conjuntos de dados utilizados na configuração experimental, e as abreviações destes conjuntos de dados podem ser encontradas na Tabela 5.

Tabela 8 – Resultados para o classificador HOMER.

		Exact Match (↑)						Accuracy (↑)						Hamming Loss (↓)					
%[1]	Db[2]	KMI[3]	KNNI[3]	MC[3]	CMC[3]	WKNNI[3]	EvoImp	KMI	KNNI	MC	CMC	WKNNI	EvoImp	KMI	KNNI	MC	CMC	WKNNI	EvoImp
5	b	43.83	48.63	49.72	48.13	49.21	**52.47**	54.05	59.51	58.92	56.42	60.03	**60.72**	06.62	05.96	05.84	06.12	06.25	**05.58**
	c	36.43	36.41	35.36	36.48	36.54	**37.93**	35.01	35.64	34.42	34.74	35.21	**35.56**	20.42	**20.21**	20.52	20.45	20.47	20.39
	e	47.12	51.04	46.72	51.12	49.37	**53.05**	51.92	54.96	50.70	55.73	53.62	**56.31**	26.64	25.62	26.43	25.28	26.41	**24.85**
	f	61.32	61.53	60.69	60.02	60.23	**63.02**	68.25	67.72	67.71	66.04	67.82	**68.42**	27.33	27.83	27.72	29.12	27.51	**27.28**
	s	51.92	54.47	50.43	51.72	55.04	**55.32**	48.78	52.66	47.95	48.83	53.12	**53.35**	14.90	14.26	15.22	14.62	14.04	**14.01**
	y	50.42	50.12	48.97	50.04	50.79	**50.92**	58.24	58.63	57.12	56.94	**58.82**	58.64	25.98	25.70	26.53	26.41	**26.02**	26.13
10	b	44.92	46.96	45.29	43.16	47.72	**48.25**	55.32	56.41	57.09	52.73	57.26	**57.74**	06.52	06.42	06.15	06.43	06.12	**06.04**
	c	35.72	36.84	35.93	36.92	36.92	**37.95**	34.33	36.38	34.62	34.95	36.03	**36.75**	20.03	19.62	19.66	19.62	19.71	**19.52**
	e	48.9	48.83	43.74	49.54	49.62	**50.64**	53.46	53.32	47.43	52.21	53.74	**53.95**	**26.21**	26.72	27.06	26.83	26.68	26.32
	f	60.74	60.32	58.93	60.74	60.48	**61.23**	68.94	71.52	67.73	67.71	**69.92**	67.87	27.42	**25.93**	27.64	27.92	27.01	27.74
	s	52.12	55.23	47.05	47.79	55.38	**55.72**	49.44	53.23	43.18	44.08	**53.56**	53.53	14.35	13.78	16.02	15.64	13.82	**13.71**
	y	50.54	50.52	50.31	48.95	50.03	**53.98**	**58.82**	58.01	58.44	57.58	57.63	58.44	25.72	25.72	**25.03**	25.51	26.21	25.87
15	b	42.13	45.62	47.24	43.32	44.68	**47.34**	50.22	56.01	57.65	51.63	56.01	**57.88**	06.83	06.11	06.07	06.82	06.14	**06.02**
	c	36.82	37.64	36.33	35.61	37.02	**37.73**	36.22	36.94	34.85	34.47	36.72	**37.23**	**18.52**	18.91	19.12	19.13	18.91	18.72
	e	47.43	47.52	42.24	48.12	45.75	**49.34**	51.92	50.11	44.42	51.58	49.59	**52.52**	27.31	26.82	26.56	25.42	26.81	**24.71**
	f	60.72	62.41	61.01	59.22	61.73	**63.98**	70.24	72.93	69.14	68.42	**73.13**	72.84	25.72	23.87	26.82	26.73	23.71	**23.62**
	s	50.91	56.42	44.45	48.42	56.41	**57.04**	47.86	54.52	40.31	45.15	54.98	**55.83**	14.72	13.61	16.02	14.47	13.42	**13.36**
	y	47.32	50.34	49.02	49.94	50.98	**51.64**	54.23	57.91	58.27	57.31	**58.82**	58.31	25.47	25.76	**24.62**	25.34	25.52	25.67
20	b	42.62	44.24	44.38	43.27	43.93	**45.14**	52.07	52.93	54.32	51.74	**55.18**	54.72	06.34	06.12	**05.74**	06.34	05.92	05.94
	c	35.72	37.34	34.92	35.34	37.25	**37.53**	34.92	36.48	34.45	34.62	36.35	**36.52**	18.24	18.46	**18.14**	18.17	18.32	18.36
	e	46.75	49.13	40.01	46.85	47.69	**49.68**	48.54	51.41	39.54	51.15	49.05	**52.37**	27.23	26.32	27.14	**25.34**	27.32	26.04
	f	58.62	60.54	56.72	62.18	60.48	**64.54**	67.82	70.31	66.97	69.11	**70.44**	69.93	27.62	25.44	27.48	25.13	25.52	**24.27**
	s	46.67	56.72	42.96	47.32	56.03	**57.54**	42.73	55.52	38.31	43.92	55.88	**56.47**	15.44	12.93	16.01	14.12	13.15	**12.62**
	y	49.07	51.62	48.57	49.16	50.82	**51.84**	57.13	**59.67**	56.78	57.12	58.46	59.44	25.65	24.72	**24.54**	24.82	25.21	24.88
25	b	43.62	41.95	43.56	40.18	41.81	**44.98**	54.43	54.34	55.82	52.53	55.78	**58.04**	05.62	05.74	05.56	05.78	05.92	**05.42**
	c	36.02	36.92	36.63	36.18	37.55	**37.79**	37.32	37.93	34.88	35.04	38.16	**38.43**	16.62	17.58	17.62	17.51	17.34	**17.35**
	e	37.34	46.42	39.28	46.32	45.17	**47.79**	35.23	46.87	37.72	47.64	47.25	**49.48**	30.53	27.82	27.01	25.87	28.23	**25.28**
	f	59.64	59.78	62.03	59.82	59.64	**63.63**	68.11	70.02	68.18	68.53	**70.35**	68.81	28.92	26.23	28.28	27.49	**25.81**	27.73
	s	47.34	55.52	41.28	46.72	55.56	**55.73**	44.01	**55.02**	38.22	44.19	54.92	55.06	14.72	**13.26**	15.32	13.64	13.38	13.39
	y	49.12	51.54	48.92	49.13	**51.52**	51.15	57.92	60.03	57.18	57.42	59.92	**60.21**	**23.88**	24.33	24.22	24.01	24.38	24.13
30	b	38.35	41.72	40.84	40.48	40.52	**42.93**	50.44	52.92	51.75	48.73	51.21	**51.68**	06.22	06.28	05.73	06.24	06.58	**05.43**
	c	35.67	36.68	36.13	35.54	36.26	**36.88**	37.55	37.74	34.98	34.32	37.05	**37.78**	**16.16**	17.04	16.46	16.87	17.22	17.06
	e	43.54	46.78	40.64	49.26	48.12	**50.04**	46.26	50.02	38.34	**53.35**	52.88	51.42	27.05	26.92	26.36	**24.34**	25.88	25.42
	f	61.14	65.26	63.54	63.02	64.12	**66.24**	71.45	74.88	72.73	74.34	74.15	**75.48**	25.72	22.76	24.94	23.96	24.02	**22.24**
	s	44.46	54.92	38.34	43.68	55.44	**56.45**	42.03	55.12	33.86	41.32	55.04	**56.25**	15.48	12.92	16.26	14.12	13.07	**12.75**
	y	49.85	52.04	48.68	50.76	52.62	**52.87**	59.92	**61.84**	59.86	60.11	61.52	61.55	23.86	23.42	22.44	**22.02**	23.36	23.24
Avg rank	-	5	2	6	4	3	**1**	4	3	6	5	2	**1**	6	2	5	3	4	**1**

[1] "%" refere-se à porcentagem de dados ausentes analisada (5%, 10%, 15%, 20%, 25% e 30%).

[2] "Db" refere-se aos conjuntos de dados utilizados na configuração experimental, e as abreviações destes conjuntos de dados podem ser encontradas na Tabela 5.

Continuando a análise dos resultados da Tabela 8, em relação à métrica ACC, o EvoImp superou os métodos de referência em 23 conjuntos de dados (63,88%). Os resultados do HL mostram que o EvoImp teve o menor erro na classificação em 19 dos 36 conjuntos de dados (52,78%). Em resumo, o EvoImp superou os métodos para todas as medidas de desempenho para o classificador HOMER, em consonância com os resultados para o classificador BR.

Multi-Label k-Nearest Neighbors

Os resultados obtidos com o classificador ML-KNN são mostrados na Tabela 9. Como pode ser observado, o EvoImp apresentou desempenho semelhante aos cenários anteriores considerando os classificadores BR e HOMER. Por exemplo, considerando a métrica principal analisada (EM), o EvoImp superou os métodos de referência em 97,22%. Considerando o ACC e o HL, o EvoImp apresentou desempenho superior para 20 (55,55%) e 22 (61,11%) conjuntos de dados, respectivamente.

Tabela 9 – Resultados para o classificador ML-KNN.

		Exact Match (↑)						**Accuracy** (↑)						**Hamming Loss** (↓)					
%[1]	**Db**[2]	**KMI**[3]	**KNNI**[3]	**MC**[3]	**CMC**[3]	**WKNNI**[3]	**EvoImp**	**KMI**	**KNNI**	**MC**	**CMC**	**WKNNI**	**EvoImp**	**KMI**	**KNNI**	**MC**	**CMC**	**WKNNI**	**EvoImp**
5	b	46.52	50.44	47.14	49.82	49.04	**51.43**	60.32	59.74	58.76	60.52	59.67	**60.76**	04.62	04.74	04.76	**04.52**	04.74	04.78
	c	36.06	35.62	36.24	36.46	36.08	**36.64**	61.22	61.40	61.59	62.26	60.61	**62.28**	13.42	13.54	**13.11**	13.23	13.52	13.24
	e	57.64	57.23	54.64	56.93	57.74	**59.32**	67.71	69.43	64.18	67.11	**69.52**	69.43	19.11	18.72	19.73	19.25	18.97	**18.35**
	f	61.42	61.61	61.25	60.98	61.82	**62.44**	70.63	70.91	**73.26**	72.42	71.21	71.54	27.72	27.31	**26.44**	26.72	27.38	27.13
	s	63.74	67.82	58.57	64.02	68.08	**68.13**	65.67	71.72	57.88	66.26	**72.15**	**72.15**	09.13	07.94	10.32	08.84	**07.92**	**07.92**
	y	56.24	57.88	55.96	55.62	58.03	**58.21**	72.46	72.92	**73.16**	72.85	72.66	72.62	19.21	**18.78**	18.82	18.94	**18.78**	**18.78**
10	b	47.02	46.04	44.28	43.93	46.07	**47.92**	55.94	**57.62**	55.53	56.24	56.88	57.04	04.62	04.64	04.66	**04.51**	04.68	04.62
	c	36.14	36.78	37.66	36.92	36.64	**37.72**	61.51	61.88	65.53	65.44	61.89	**65.66**	13.01	13.22	**12.46**	12.52	13.21	**12.46**
	e	55.46	55.18	50.82	56.06	54.07	**56.96**	65.62	64.05	56.69	65.92	64.06	**66.67**	20.26	20.62	21.87	19.72	20.96	**19.62**
	f	59.94	60.26	59.42	62.37	59.48	**63.11**	72.85	72.48	70.82	73.64	72.05	**73.86**	26.42	26.78	27.61	25.12	27.04	**24.92**
	s	60.92	69.46	48.92	59.88	69.72	**69.87**	62.11	74.82	43.89	61.34	**74.96**	**74.96**	09.52	07.38	11.92	09.56	**07.32**	**07.32**
	y	56.23	58.14	54.29	54.92	58.14	**58.46**	73.02	73.24	**73.38**	73.02	73.16	73.22	18.78	18.22	18.78	18.46	18.24	**18.12**
15	b	45.62	45.34	45.46	46.01	43.54	**48.16**	56.24	57.58	55.24	57.26	56.34	**60.36**	04.51	04.42	04.54	**04.36**	04.52	**04.36**
	c	36.52	37.74	**39.56**	39.02	37.74	**39.56**	62.22	61.01	67.28	67.09	61.46	**67.47**	12.62	12.74	**11.64**	11.72	12.76	**11.64**
	e	55.02	54.64	47.48	53.29	54.24	**55.31**	64.44	65.45	54.78	64.49	65.01	**65.86**	20.62	19.88	21.12	19.91	20.26	**19.52**
	f	60.56	61.14	56.62	58.48	60.62	**61.87**	**72.96**	72.02	72.14	70.32	71.75	72.42	**26.16**	26.62	27.34	28.18	27.12	26.41
	s	59.56	70.24	42.32	57.51	**70.56**	**70.56**	59.82	75.14	35.72	56.62	**75.38**	75.28	09.92	**07.11**	12.72	09.87	**07.11**	**07.11**
	y	53.74	59.82	55.81	55.66	59.62	**59.92**	69.65	73.19	73.81	**74.36**	73.02	73.27	19.29	**17.44**	17.92	17.87	17.56	**17.44**
20	b	43.82	45.62	45.04	46.49	44.52	**50.91**	57.56	57.14	54.97	55.62	56.65	**57.64**	**04.14**	**04.14**	04.22	**04.14**	**04.14**	04.71
	c	38.29	37.82	**40.14**	40.02	37.96	**40.14**	63.76	62.92	69.94	**70.08**	62.94	69.96	11.81	12.44	**11.05**	**11.05**	12.41	**11.05**
	e	53.42	55.14	41.06	52.07	55.26	**55.96**	59.41	63.97	42.25	61.26	64.84	**65.52**	21.21	20.18	22.79	**19.32**	20.31	20.26
	f	61.71	61.67	59.49	60.62	61.82	**63.17**	74.36	75.34	74.82	74.76	74.87	**76.26**	24.15	24.62	25.84	25.76	24.74	**23.92**
	s	54.15	71.92	39.88	54.59	71.41	**72.06**	51.82	**77.34**	34.32	52.56	77.22	77.26	10.31	**06.72**	12.56	10.01	**06.72**	**06.72**
	y	54.66	60.24	53.32	53.84	60.62	**60.76**	68.82	74.74	**76.06**	75.92	74.94	75.06	19.88	16.82	17.76	17.51	**16.72**	17.44
25	b	44.16	44.24	41.36	42.21	47.06	**47.37**	56.72	58.22	56.86	56.81	58.56	**58.82**	**03.82**	03.86	03.84	03.85	**03.82**	**03.82**
	c	39.28	39.16	**41.94**	41.25	38.79	**41.94**	63.32	62.92	70.64	69.62	62.66	**70.71**	11.52	12.16	**10.34**	10.55	12.18	**10.34**
	e	43.54	53.02	33.78	47.92	50.52	**53.64**	42.92	58.48	30.36	54.64	55.57	**59.16**	21.92	21.54	22.72	**19.74**	22.02	21.24
	f	60.56	60.32	62.24	58.76	59.84	**62.55**	75.42	75.29	74.21	**75.82**	74.95	74.36	26.39	**25.02**	25.96	26.52	25.84	25.64
	s	52.65	72.24	36.32	50.96	72.34	**72.58**	49.64	78.31	32.02	48.36	78.17	**78.46**	10.52	**06.41**	12.44	10.42	**06.41**	**06.41**
	y	54.42	61.25	54.49	54.31	61.06	**61.88**	76.76	74.91	77.14	**77.36**	75.05	75.21	17.02	**16.16**	16.82	16.91	**16.16**	16.78
30	b	42.32	42.16	38.58	42.57	39.82	**46.99**	55.11	54.46	55.34	55.26	54.25	**55.76**	**03.92**	04.01	**03.92**	**03.92**	04.08	04.06
	c	39.74	38.26	42.52	42.64	38.12	**42.71**	65.76	63.84	74.08	**73.65**	64.24	**73.65**	10.92	11.86	**09.68**	09.72	11.74	09.78
	e	50.02	52.99	31.62	52.64	54.94	**55.08**	57.56	63.01	28.56	64.88	64.86	**65.57**	20.91	20.35	22.46	19.62	19.68	**19.51**
	f	**65.72**	63.36	61.38	64.32	60.71	65.19	**79.12**	77.45	81.27	77.99	77.32	76.06	**22.61**	24.22	23.45	22.98	25.16	24.12
	s	51.32	72.26	34.62	48.84	72.26	**72.32**	49.44	78.92	31.56	45.68	**79.12**	79.04	10.62	06.26	12.18	10.44	**06.12**	**06.12**
	y	52.56	62.58	53.72	54.31	62.88	**63.09**	73.61	75.85	**79.42**	79.24	76.16	76.12	18.88	15.26	16.12	16.14	**15.14**	**15.14**
Avg rank	-	5	2	6	4	3	**1**	6	3	5	2	4	**1**	5	2	6	3	4	**1**

[1] "%" refere-se à porcentagem de dados ausentes analisada (5%, 10%, 15%, 20%, 25% e 30%).

[2] "Db" refere-se aos conjuntos de dados utilizados na configuração experimental, e as abreviações destes conjuntos de dados podem ser encontradas na Tabela 5.

Classifier Chains

Os resultados para o *Classifier Chains* são apresentados na Tabela 10. Novamente, o EvoImp superou os métodos de referência para todas as medidas de avaliação consideradas:

EM com superioridade em 32 dos 36 conjuntos de dados (88.88%), ACC com 30 (83.33%), e HL com 22 (61.11%).

Tabela 10 – Resultados para o *Classifier Chains*.

		Exact Match (↑)						**Accuracy** (↑)						**Hamming Loss** (↓)					
%[1]	**Db**[2]	**KMI**[3]	**KNNI**[3]	**MC**[3]	**CMC**[3]	**WKNNI**[3]	**EvoImp**	**KMI**	**KNNI**	**MC**	**CMC**	**WKNNI**	**EvoImp**	**KMI**	**KNNI**	**MC**	**CMC**	**WKNNI**	**EvoImp**
5	b	49.62	48.56	49.91	50.45	50.21	**51.46**	61.52	60.81	61.44	60.99	61.62	**62.26**	05.21	05.25	05.06	05.11	05.36	**05.02**
	c	34.64	34.92	34.34	34.56	34.81	**35.34**	40.06	40.14	40.21	**40.46**	40.05	40.32	17.34	17.46	17.14	**17.01**	17.42	17.13
	e	49.82	49.91	45.86	48.82	48.81	**52.06**	55.45	55.68	52.82	53.59	54.76	**56.56**	25.72	**25.42**	27.34	26.72	25.86	25.61
	f	59.52	58.98	60.56	59.76	59.32	**62.26**	66.72	67.11	68.62	68.18	67.92	**70.06**	29.71	29.82	28.26	28.71	29.08	**26.86**
	s	56.52	59.78	51.51	54.26	58.17	**59.96**	56.82	61.84	51.42	55.46	60.36	**62.02**	14.14	13.62	15.52	13.88	14.04	**13.52**
	y	49.29	49.82	48.14	48.12	49.98	**50.36**	56.22	56.38	54.54	55.02	56.44	**56.48**	26.56	26.48	26.61	26.46	**26.42**	26.51
10	b	46.22	48.16	48.44	45.12	45.92	**49.56**	57.61	60.52	60.38	55.76	58.13	**61.62**	05.41	05.36	05.02	05.38	05.56	**04.92**
	c	**36.18**	35.56	34.82	35.08	35.31	36.06	**45.45**	42.02	40.76	40.61	41.68	41.82	**16.32**	16.52	16.31	16.36	16.72	16.56
	e	45.92	49.48	41.71	49.66	48.38	**50.92**	52.36	54.25	46.07	52.02	**55.06**	54.92	28.08	26.89	29.52	27.24	26.56	**26.44**
	f	57.16	59.62	57.74	58.33	61.05	**61.58**	69.72	71.76	71.41	68.12	72.26	**72.82**	28.18	27.12	26.76	28.81	26.08	**25.36**
	s	53.54	59.92	48.46	52.72	58.71	**60.08**	54.16	61.85	46.32	51.16	60.32	**61.96**	14.12	13.18	15.77	14.09	13.66	**13.12**
	y	48.52	49.26	47.24	46.96	49.32	**50.56**	55.34	55.69	53.82	53.89	56.92	**57.75**	26.24	26.62	26.54	26.26	26.28	**25.66**
15	b	44.52	45.38	**46.62**	41.91	44.28	**46.62**	55.16	56.21	**57.62**	52.88	55.62	57.61	05.72	05.32	05.08	05.59	05.32	**05.06**
	c	**35.82**	35.06	35.39	34.62	34.51	35.75	41.46	41.92	41.18	40.92	41.36	**42.52**	16.06	16.28	**15.62**	15.76	16.38	15.67
	e	47.96	47.92	42.16	50.31	47.88	**50.96**	50.92	**56.28**	42.06	55.24	53.76	56.08	27.12	26.16	27.42	24.54	26.76	**24.18**
	f	61.52	62.36	58.11	60.89	62.82	**64.89**	72.02	73.46	72.09	72.92	73.74	**74.19**	26.12	24.87	26.42	25.74	24.32	**23.62**
	s	51.72	59.86	47.31	50.26	59.02	**60.18**	52.04	62.42	44.26	48.61	61.02	**62.75**	14.56	12.92	15.24	13.56	13.12	**12.82**
	y	46.11	51.36	48.34	46.72	51.06	**51.82**	53.16	57.91	55.22	53.88	57.46	**58.16**	26.02	25.05	25.14	26.02	25.38	**24.96**
20	b	42.02	44.84	43.96	43.46	44.33	**46.17**	53.25	57.48	55.44	53.13	55.42	**58.14**	05.52	05.06	05.07	05.32	05.26	**04.92**
	c	35.74	36.16	34.91	34.78	35.26	**36.62**	41.26	42.34	41.85	40.38	41.34	**43.48**	15.46	15.82	**14.84**	15.21	16.16	15.32
	e	44.98	47.93	38.52	47.94	45.92	**49.38**	48.92	51.91	36.74	52.58	50.03	**53.55**	27.82	26.68	28.39	**24.91**	27.64	25.86
	f	57.62	61.38	58.97	60.76	59.02	**61.88**	68.91	70.86	70.72	70.54	**71.26**	**71.26**	28.92	**25.51**	26.75	26.62	25.66	25.57
	s	46.52	**59.77**	42.26	48.05	59.18	59.76	44.22	61.64	38.38	45.72	**61.62**	**61.62**	15.12	**12.64**	16.22	13.79	12.72	12.69
	y	48.82	49.91	46.06	45.92	51.38	**51.41**	56.42	56.56	53.08	53.16	58.51	**58.68**	25.66	25.88	25.62	25.64	24.82	**24.71**
25	b	44.13	43.42	43.15	41.28	45.42	**45.78**	54.62	55.51	55.06	51.62	58.34	**58.42**	05.06	05.21	04.75	04.92	04.74	**04.72**
	c	**37.56**	35.88	36.12	35.51	36.36	37.23	43.72	41.91	43.26	42.72	42.81	**44.86**	14.64	15.72	**14.24**	14.28	15.42	14.56
	e	42.24	46.12	37.26	45.08	46.22	**48.76**	43.02	49.81	35.92	48.76	49.31	**52.19**	26.82	27.18	27.26	27.61	27.15	**26.22**
	f	59.32	60.66	60.92	59.28	61.24	**62.35**	**72.66**	72.32	72.44	71.26	72.58	72.22	26.87	25.92	26.84	26.92	25.65	**25.61**
	s	46.75	59.94	39.62	45.96	60.12	**61.56**	43.94	63.26	36.72	42.92	62.26	**63.98**	14.65	12.34	15.82	13.76	12.52	**11.77**
	y	46.15	51.52	46.63	46.28	51.39	**51.67**	54.12	58.72	54.55	53.59	58.92	**58.97**	24.36	24.52	**24.24**	24.52	24.56	24.48
30	b	42.06	40.62	44.97	40.66	41.34	**45.49**	54.25	52.87	55.92	51.46	52.85	**56.49**	05.02	05.28	04.76	05.11	05.12	**04.64**
	c	36.83	35.05	35.42	36.58	34.92	**38.76**	44.24	41.92	43.76	44.25	40.92	**45.81**	14.02	15.26	**13.43**	13.49	15.44	13.82
	e	42.55	44.82	38.18	48.09	45.74	**48.80**	47.06	50.22	37.79	52.95	50.91	**54.42**	26.90	27.44	26.66	24.54	27.48	**24.26**
	f	61.60	61.65	59.86	62.92	63.31	**64.14**	75.42	73.26	75.18	74.82	74.19	**75.47**	24.62	26.06	24.98	24.76	25.12	**24.14**
	s	43.40	59.85	38.73	45.44	58.83	**60.15**	41.62	62.53	34.64	43.23	62.18	**63.29**	15.20	**12.02**	15.36	13.21	12.25	12.03
	y	49.77	52.86	47.40	47.27	52.56	**53.12**	57.91	60.48	54.96	55.64	60.32	**60.65**	24.10	23.23	23.35	**23.01**	23.46	23.12
Avg rank	-	4	2	6	5	3	**1**	4	2	5	6	3	**1**	6	3	4	2	5	**1**

[1] "%" refere-se à porcentagem de dados ausentes analisada (5%, 10%, 15%, 20%, 25% e 30%).

[2] "Db" refere-se aos conjuntos de dados utilizados na configuração experimental, e as abreviações destes conjuntos de dados podem ser encontradas na Tabela 5.

Ensemble of Classifier Chains

O último cenário analisado considerou o método de *Ensemble of Classifier Chains*. Os resultados estão apresentados na Tabela 11. Os resultados obtidos com o ECC (Tabela 11) também mostram uma vantagem significativa do EvoImp sobre os concorrentes nas análises realizadas. No entanto, o EvoImp teve a menor performance, com superioridade numérica em 29 (80.55%) conjuntos de dados na avaliação com EM, 16 (44.44%) para ACC e 17 (47.22%) para HL.

Tabela 11 – Resultados para o classificador *Ensemble of Classifier Chains*.

		Exact Match (↑)						**Accuracy** (↑)						**Hamming Loss** (↓)					
%[1]	**Db**[2]	**KMI**[3]	**KNNI**[3]	**MC**[3]	**CMC**[3]	**WKNNI**[3]	**EvoImp**	**KMI**	**KNNI**	**MC**	**CMC**	**WKNNI**	**EvoImp**	**KMI**	**KNNI**	**MC**	**CMC**	**WKNNI**	**EvoImp**
5	b	53.72	54.43	52.75	52.79	53.61	**54.69**	64.42	65.78	64.31	63.40	**65.76**	65.32	04.23	04.26	04.27	**04.21**	04.26	04.28
	c	54.52	54.88	54.29	54.12	**55.16**	36.64	54.36	55.02	**56.89**	55.72	54.64	56.09	14.12	14.16	13.79	13.82	14.16	**13.72**
	e	55.56	56.82	52.16	58.26	56.92	**58.68**	61.92	**65.86**	59.12	64.57	63.42	63.93	20.22	19.05	20.93	**18.94**	19.76	19.18
	f	61.32	62.09	61.60	62.62	62.25	**65.68**	71.02	70.26	70.72	71.58	69.72	**72.56**	25.71	25.46	26.22	25.67	25.92	**24.84**
	s	59.92	63.93	56.14	57.92	**64.46**	**64.46**	59.82	65.84	54.83	57.32	65.84	**65.89**	09.82	**08.83**	10.30	09.82	08.87	08.88
	y	54.52	54.86	54.24	54.12	55.18	**55.39**	69.02	68.80	**69.44**	69.41	69.13	69.39	20.52	20.46	19.91	**19.82**	20.36	20.23
10	b	45.25	46.62	49.13	47.99	48.10	**49.28**	60.02	**61.97**	61.32	58.92	61.65	61.26	04.42	04.41	**04.23**	04.35	04.36	04.27
	c	54.16	**55.82**	51.96	52.98	55.72	37.51	56.16	55.54	57.22	57.58	55.46	**58.14**	13.55	13.83	**13.14**	13.18	13.86	13.33
	e	53.82	53.87	46.85	54.20	54.51	**55.35**	60.83	60.71	52.02	**61.93**	60.62	61.14	21.23	**20.92**	22.46	20.32	21.27	**20.92**
	f	61.33	61.37	61.52	62.78	60.93	**65.03**	71.51	72.28	72.92	73.33	73.09	**74.05**	26.24	25.52	**24.46**	24.84	25.82	24.83
	s	59.05	64.82	52.73	56.77	**65.26**	**65.26**	58.14	66.28	48.72	55.16	**66.87**	**66.87**	09.82	08.74	10.93	09.78	**08.42**	**08.42**
	y	54.19	55.82	51.97	52.93	54.08	**56.06**	70.07	70.32	69.49	69.72	70.17	**70.66**	19.92	19.78	19.91	19.82	19.79	**19.63**
15	b	47.24	47.39	47.82	44.18	47.33	**50.04**	58.07	**61.09**	60.81	57.66	60.48	59.64	04.41	**04.23**	**04.23**	04.37	04.28	04.25
	c	51.92	**56.09**	52.90	53.77	56.06	38.25	57.78	57.62	**60.23**	58.54	57.93	60.12	12.96	13.25	**12.37**	12.53	13.12	12.39
	e	53.02	52.78	45.13	54.54	52.02	**55.34**	59.75	**63.78**	48.52	62.46	61.01	62.49	21.35	20.03	22.67	19.68	20.72	**19.29**
	f	62.52	63.89	59.67	60.45	64.32	**67.09**	73.74	73.05	71.20	71.36	73.72	**74.48**	24.34	23.95	26.57	26.02	23.69	**22.92**
	s	56.64	64.82	49.73	53.68	64.62	**64.88**	56.32	65.56	46.72	51.20	**66.06**	65.52	10.17	**08.52**	10.96	10.04	08.69	**08.52**
	y	51.94	56.01	52.92	53.78	56.09	**56.32**	66.46	70.52	71.58	71.05	70.06	**72.08**	20.02	19.39	**19.13**	18.87	19.39	19.16
20	b	47.15	46.53	46.58	46.92	47.59	**51.07**	56.92	59.26	58.68	58.22	58.47	**59.46**	04.15	04.16	04.02	04.08	04.19	**03.96**
	c	53.82	56.36	52.78	52.46	**57.17**	39.12	58.82	57.67	61.92	62.09	57.35	**62.51**	12.26	12.88	11.66	11.64	12.98	**11.62**
	e	51.32	53.28	45.56	51.98	52.40	**54.67**	57.16	60.01	43.78	59.32	60.92	**61.16**	20.74	20.42	22.29	**19.44**	20.82	20.07
	f	59.92	62.56	61.07	61.74	62.02	**63.89**	71.45	72.49	71.63	70.67	72.42	**73.46**	26.57	24.09	25.55	26.79	24.34	**23.34**
	s	49.85	65.32	46.39	53.87	65.26	**65.37**	47.12	66.96	41.75	51.90	66.88	**66.95**	11.12	**08.23**	11.54	09.56	08.35	**08.23**
	y	53.88	56.32	52.70	52.45	57.19	**57.42**	69.36	70.97	**72.62**	72.06	70.97	71.24	19.82	19.09	**18.52**	18.67	18.92	18.80
25	b	44.15	43.63	45.42	45.77	44.26	**48.56**	59.72	**59.82**	58.79	58.36	59.54	59.55	03.72	03.77	03.76	03.75	03.82	**03.65**
	c	52.92	**57.29**	53.43	53.14	57.16	40.32	61.26	59.18	**64.93**	63.64	59.72	63.56	11.68	12.46	10.92	**10.14**	12.46	11.24
	e	44.38	51.93	38.00	50.97	50.42	**53.83**	46.62	56.54	38.08	58.62	55.46	**61.24**	22.02	22.06	22.28	18.42	22.37	**18.24**
	f	61.54	61.90	61.98	60.86	60.82	**63.44**	73.02	**74.30**	73.56	73.24	72.38	73.11	25.88	**24.02**	25.56	25.14	25.21	25.48
	s	51.36	66.55	44.27	51.42	65.76	**66.67**	48.65	68.52	40.28	50.07	68.02	**68.79**	10.52	**07.94**	11.56	09.62	08.17	07.99
	y	52.92	57.28	53.44	53.15	57.16	**57.72**	73.07	71.45	**73.42**	72.84	71.96	72.02	17.88	18.74	**17.73**	18.05	18.42	18.27
30	b	42.35	42.28	46.43	47.35	42.02	**49.86**	55.14	54.92	54.94	**58.26**	55.07	56.51	03.95	04.17	03.92	**03.86**	04.12	03.88
	c	54.82	**58.26**	53.42	53.47	57.89	40.80	63.32	58.67	**66.69**	66.12	59.77	66.26	11.02	12.28	10.24	**10.32**	12.11	10.39
	e	49.34	53.72	35.23	54.67	54.15	**55.16**	56.01	62.98	35.36	**65.25**	63.38	64.67	21.42	19.72	22.10	18.55	19.76	**18.22**
	f	**65.73**	63.65	62.47	64.82	62.89	65.64	**78.23**	75.35	76.48	76.72	74.86	77.47	22.76	23.27	24.26	22.82	24.07	**22.52**
	s	48.96	65.70	42.77	50.54	65.02	**65.76**	47.37	68.92	38.75	48.69	67.52	**68.97**	10.92	**07.65**	11.32	09.67	08.02	**07.65**
	y	54.82	58.28	53.49	53.42	57.87	**58.34**	72.92	73.19	74.92	**75.28**	73.42	73.46	18.28	17.71	16.92	**16.87**	17.76	17.62
Avg rank	-	5	2	6	4	3	**1**	6	2	5	3	4	**1**	6	3	4	2	5	**1**

[1] "%" refere-se à porcentagem de dados ausentes analisada (5%, 10%, 15%, 20%, 25% e 30%).

[2] "Db" refere-se aos conjuntos de dados utilizados na configuração experimental, e as abreviações destes conjuntos de dados podem ser encontradas na Tabela 5.

Em resumo, o desempenho do EvoImp para o ECC apresenta o mesmo padrão descrito nos cenários anteriores, demonstrando a robustez do método.

6.2 DISCUSSÃO

Considerando os cenários e as bases de dados utilizadas, o EvoImp mostrou ser competitivo em todos os cenários de classificação, destacando o fato de que a otimização da imputação por meio de estratégias evolutivas, como algoritmos genéticos, é uma excelente alternativa para lidar com valores ausentes na fase de pré-processamento da análise de dados. Deve-se observar que o algoritmo criado realizou otimizações com base em métodos simples de imputação (aplicados à população inicial do EvoImp). Considerando os experimentos computacionais, outros fatores devem ser destacados em relação ao desempenho do EvoImp. Esses destaques são feitos a seguir.

Maximização na corretude dos rótulos

O principal objetivo da classificação, especialmente neste estudo, é a rotulação correta das instâncias de dados, uma tarefa que está se tornando cada vez mais complexa no cenário de rotulação múltipla. Na medida EM, onde o classificador deve rotular

corretamente todas as classes de uma instância para que possam ser contadas corretamente, o método proposto obteve melhor desempenho em 92,22% de todos os conjuntos de dados em todos os cenários. Esse desempenho é mais evidente em BR, HOMER e ML-KNN, com 35 dos 36 conjuntos de dados. Outra medida que permite essa conclusão é a ACC. O desempenho superior alcançado pelo EvoImp é mais aparente nas análises com os classificadores CC e HOMER (com 30 e 23 conjuntos de dados, respectivamente).

Em termos gerais, o EvoImp foi melhor em 68,3% de todos os conjuntos de dados utilizados. Isso pode ser explicado pelo fato de que esta medida é flexível em relação ao número de sucessos alcançados pelos rótulos. Por exemplo, se uma instância pertence a cinco rótulos e obtém quatro rótulos corretos, ela alcança 80% de precisão. Ao mesmo tempo, o excelente desempenho do ACC indica que o classificador pode aumentar sua capacidade de rotulagem. Isso pode ser confirmado analisando o erro de classificação avaliado usando HL. Nesta métrica, o método proposto obteve o menor erro (53,33%).

Vale ressaltar que os resultados obtidos refletem a ordem lexicográfica escolhida (como explicado na subseção "Função de Aptidão" 4.3), demonstrando a superioridade do método sobre todos os outros. Uma comparação mostra que, à medida que o ACC aumenta, há uma redução automática no erro do HL, justificando o uso da ordem lexicográfica em vez de abordagens mais complexas, como a Análise da Fronteira de Pareto, usada para lidar com medidas conflitantes.

Desempenho superior em conjuntos de dados de diferentes domínios e tamanhos

Os seis conjuntos de dados utilizados nos experimentos podem ser divididos em termos de:

- domínios diferentes - os conjuntos de dados de rótulo múltiplo utilizados estavam relacionados às áreas de:
  - áudio (1);
  - música (2);
  - imagem (2); e
  - biologia (1).
- seus tamanhos - considerando o número de instâncias e atributos, como feito por (SCHMITT; MANDEL; GUEDJ, 2015).

Esses conjuntos de dados foram elaborados para fornecer uma configuração experimental robusta, simulando problemas do mundo real diversos. Observou-se que o EvoImp teve desempenho superior em todos os testes, provando que o método é robusto em conjuntos de dados de diferentes domínios e tamanhos.

Desempenho estável nas taxas de distribuição de valores ausentes sob estudo

Uma avaliação crítica deste estudo está relacionada à relação entre a porcentagem de valores ausentes e as medidas de desempenho. Os resultados mostram que o EvoImp mantém sua consistência, mesmo com variações, que, neste estudo, foram entre 5% e 30% (com uma taxa de k = 5%).

Essas taxas concordam com as usadas na maioria dos estudos na literatura - um trabalho relacionado que aborda essa discussão é Chiu et al. (2022). Um total de 48 artigos relacionados de 2011 a 2021 foram selecionados nesta investigação. Sobre as taxas de valores ausentes, esta revisão indicou que 60,4% utilizaram taxas $\leq 30\%$ ou não revelaram suas taxas de ausência para a experimentação.

Os aspectos supracitados demonstram que o EvoImp é adequado para tratamento de valores ausentes em cenários do mundo real.

7 CONCLUSÃO E TRABALHOS FUTUROS

Neste capítulo final são realizadas as reflexões finais do trabalho ressaltando as contribuições obtidas. Além disso, são destacados os trabalhos futuros que irão ser realizados como fruto da pesquisa e, por fim, são apresentadas as produções bibliográficas e técnicas geradas.

7.1 CONSIDERAÇÕES FINAIS

As análises de dados conduzidas em conjuntos de dados do mundo real deixam claro que há uma necessidade crítica de lidar com valores ausentes no domínio da classificação multirrótulo. A presença ubíqua de VAs e o fato de que a maioria das técnicas empregadas só funcionam ou garante bom desempenho quando aplicadas a conjuntos de dados com casos completos destacam a necessidade de enfrentar esse problema. Métodos de imputação de dados surgiram como uma solução alternativa, buscando valores plausíveis para preencher os ausentes.

Portanto, foi proposto neste estudo o EvoImp, um método de imputação baseado em algoritmos genéticos para a otimização de múltiplas imputações para dados ausentes aplicados à aprendizagem multirrótulo. Para validação, o método foi submetido a um extenso processo de *benchmarking* experimental com vários conjuntos de dados multirrótulo e comparado com outros métodos de imputação de ponta. Seis taxas de valores ausentes foram aplicadas aos conjuntos de dados para simular o mecanismo MCAR. Os resultados foram analisados usando cinco classificadores: *Binary Relevance*, *Hierarchy of Multi-label Classifier*, *Multi-Label k-Nearest Neighbors*, *Classifier Chains* e *Ensembles of Classifier Chains*. Três medidas de avaliação bem conhecidas foram adotadas para avaliar os experimentos: *Exact Match*, *Accuracy* e *Hamming-loss*.

O EvoImp obteve resultados consistentemente superiores nos cenários avaliados, destacando-se quantitativamente em relação aos demais métodos. Esses resultados permitem concluir que o método proposto é adequado para aplicação em cenários do mundo real. Além de uma abordagem inovadora para lidar com valores ausentes na classificação multirrótulo, o trabalho presente contribui para o corpo de conhecimento por:

1. Avaliar o impacto de dados ausentes na classificação multirrótulo para melhorar a robustez da classificação;

2. Fornecer uma extensa comparação experimental de muitos algoritmos de imputação

de dados de ponta, classificadores de aprendizado de máquina multirrótulo e medidas de desempenho;

3. Disponibilizar códigos-fonte e resultados de experimentos em um repositório do GitHub.

7.2 TRABALHOS FUTUROS

Em trabalhos futuros, pretende-se avaliar outros mecanismos de ausência além do MCAR e ajustar o método para lidar com altas taxas de dados ausentes ($> 30\%$). Experimentos também poderiam ser realizados para fazer o EvoImp aprender seus parâmetros (Adaptativo).

Outra possibilidade é realizar a tradução e validação da versão do método na linguagem *Python* e publicação como pacotes *Python* (pip) no *PyPi*[1], bem como verificar com métodos estatísticos os casos em que o método não obtiver os melhores resultados.

Pretende-se, por meio de metodologias como a *"umbrella review"* (AROMATARIS et al., 2015; FUSAR-POLI; RADUA, 2018) realizar uma integração sobre o tema de dados ausentes, a partir das revisões e levantamentos sistemáticos da literatura (como os citados durante essa tese (LIN; TSAI, 2020; CHIU et al., 2022; REN et al., 2023; NUGROHO; SURENDRO, 2024)).

Além disso, pode-se testar o método no tratamento de *Missing Labels* em datasets MLC e em diferentes contextos, tais como IoT e Medicina. Também, pretende-se de investigar a influência das características de cardinalidade e densidade na aprendizagem multirrótulo com valores ausentes. Com relação a outros métodos a serem utilizados, conforme apontado na análise de (NUGROHO; SURENDRO, 2024), a utilização de *deep learning* como um método de imputação ainda deve ser melhor explorado.

7.3 PUBLICAÇÕES RELACIONADAS A PESQUISA

PRODUÇÃO BIBLIOGRÁFICA

O artigo publicado possui classificação A1 em Engenharias IV conforme estrato Qualis CAPES (2017-2020). Uma cópia do artigo encontra-se no Anexo I.

- Jacob Junior AFL, do Carmo FA, de Santana AL, Santana EEC, Lobato FMF (2024) EvoImp: Multiple Imputation of Multi-label Classification data with a genetic algorithm. PLOS ONE 19(1): e0297147. DOI: 10.1371/journal.pone.0297147

[1] https://pypi.org/

PRODUÇÕES TÉCNICAS

Os seguintes produtos técnicos foram gerados e disponibilizados em plataformas bem aceitas pela comunidade. Nesse caso, vale destacar que esses tipos de produto tem por finalidade viabilizar uma melhor replicabilidade dos experimentos realizados.

- Antonio F. L. Jacob Jr., Fabrício A. do Carmo, Ádamo L. de Santana, Ewaldo Santana, & Fábio M. F. Lobato. (2023). Multi-Label Datasets with Missing Values [Data set]. Zenodo. DOI: 10.5281/zenodo.7748933
- Antonio F. L. Jacob Jr., Fabrício A. do Carmo, Ádamo L. de Santana, Ewaldo Santana, & Fábio M. F. Lobato. "EvoImp: Algoritmo Genético para Imputação Múltipla no contexto de Classificação Multirótulo". Patente: Programa de Computador. Número do registro: BR512024000513-7, data de registro: 19/01/2024, Instituição de registro: INPI - Instituto Nacional da Propriedade Industrial e, também, o repositório digital dos códigos-fontes sob a licença "MIT License": Github.

Referências

ABDI, Y.; ASADPOUR, M.; SEYFARI, Y. μmosm: A hybrid multi-objective micro evolutionary algorithm. *Engineering Applications of Artificial Intelligence*, Elsevier, v. 126, p. 107000, 2023. Citado na página 17.

ABOU-FOUL, M.; RUIZ-ALBA, J. L.; LÓPEZ-TENORIO, P. J. The impact of artificial intelligence capabilities on servitization: The moderating role of absorptive capacity-a dynamic capabilities perspective. *Journal of Business Research*, Elsevier, v. 157, p. 113609, 2023. Citado na página 15.

ADHIKARI, D.; JIANG, W.; ZHAN, J.; HE, Z.; RAWAT, D.; AICKELIN, U.; KHORSHIDI, H. A comprehensive survey on imputation of missing data in internet of things. *ACM Computing Surveys*, v. 55, n. 7, p. 1–38, Dec 15 2022. Citado na página 16.

AHA, D. W. Special issue on lazy learning. *Artificial Intelligence Review*, v. 11, p. 7–10, 1997. Citado na página 31.

ALMEIDA, M. M. et al. *Imputação de dados faltosos em séries Temporais Univariadas utilizando meta-aprendizado baseado em Rede Neural LSTM Híbrida*. 2023. Citado na página 22.

APPELBAUM, S.; KRÜERKE, D.; BAUMGARTNER, S.; SCHENKER, M.; OSTERMANN, T. Development, implementation and validation of a stochastic prediction model of uicc stages for missing values in large data sets in a hospital cancer registry. In: *HEALTHINF*. [S.l.: s.n.], 2023. p. 117–123. Citado na página 15.

AROMATARIS, E.; FERNANDEZ, R.; GODFREY, C. M.; HOLLY, C.; KHALIL, H.; TUNGPUNKOM, P. Summarizing systematic reviews: methodological development, conduct and reporting of an umbrella review approach. *JBI Evidence Implementation*, LWW, v. 13, n. 3, p. 132–140, 2015. Citado na página 58.

ARRIETA, A. B.; DÍAZ-RODRÍGUEZ, N.; SER, J. D.; BENNETOT, A.; TABIK, S.; BARBADO, A.; GARCÍA, S.; GIL-LÓPEZ, S.; MOLINA, D.; BENJAMINS, R. et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, Elsevier, v. 58, p. 82–115, 2020. Citado na página 15.

BAI, R.; CHEN, X.; CHEN, Z.-L.; CUI, T.; GONG, S.; HE, W.; JIANG, X.; JIN, H.; JIN, J.; KENDALL, G. et al. Analytics and machine learning in vehicle routing research. *International Journal of Production Research*, Taylor & Francis, v. 61, n. 1, p. 4–30, 2023. Citado na página 15.

BANERJEE, A.; GHOSH, J. Scalable clustering algorithms with balancing constraints. *Data Min Knowl Disc*, v. 13, p. 365–395, 2006. Citado 2 vezes nas páginas 30 e 49.

BATISTA, G. E.; MONARD, M. C. et al. A study of k-nearest neighbour as an imputation method. *His*, v. 87, n. 251-260, p. 48, 2002. Citado na página 25.

BEZERRA, E. D. C.; TELES, A. S.; COUTINHO, L. R.; SILVA, F. J. da Silva e. Dempster–shafer theory for modeling and treating uncertainty in iot applications based on complex event processing. *Sensors*, v. 21, n. 5, 2021. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/21/5/1863>. Citado na página 21.

BLEIDORN, M. T.; PINTO, W. d. P.; SCHMIDT, I. M.; MENDONÇA, A. S. F.; REIS, J. A. T. d. Abordagens metodológicas para imputação de dados faltantes de vazões médias mensais. *Revista Ambiente & Água*, SciELO Brasil, v. 17, p. e2795, 2022. Citado na página 22.

BOGATINOVSKI, J.; TODOROVSKI, L.; DZEROSKI, S.; KOCEV, D. Comprehensive comparative study of multi-label classification methods. *Expert Syst Appl*, v. 203, p. 117215, 2022. Citado na página 49.

BOKRANTZ, J.; SUBRAMANIYAN, M.; SKOOGH, A. Realising the promises of artificial intelligence in manufacturing by enhancing crisp-dm. *Production Planning & Control*, Taylor & Francis, p. 1–21, 2023. Citado na página 22.

BRZOZOWSKA, J.; PIZOŃ, J.; BAYTIKENOVA, G.; ARKADIUSZ, G.; ZAKIMOVA, A.; PIOTROWSKA, K. Data engineering in crisp-dm process production data–case study. *Applied Computer Science*, v. 19, n. 3, p. 83–95, 2023. Citado na página 22.

CHENG, Y.; SONG, F.; QIAN, K. Missing multi-label learning with non-equilibrium based on two-level autoencoder. *Applied Intelligence*, p. 1–9, Oct 1 2021. Citado 2 vezes nas páginas 35 e 36.

CHIU, P.; SELAMAT, A.; KREJCAR, O.; KUOK, K.; BUJANG, S.; FUJITA, H. Missing value imputation designs and methods of nature-inspired metaheuristic techniques: A systematic review. *IEEE Access*, p. 61544–61566, May 9 2022. Citado 6 vezes nas páginas 16, 29, 34, 46, 56 e 58.

CIRQUEIRA, D.; ALMEIDA, F.; CAKIR, G.; JACOB, A.; LOBATO, F.; BEZBRADICA, M.; HELFERT, M. Explainable sentiment analysis application for social media crisis management in retail. ScitePress, 2020. Citado na página 15.

CIRQUEIRA, D.; PINHEIRO, M. F.; JR., A. J.; LOBATO, F.; SANTANA, A. A literature review in preprocessing for sentiment analysis for brazilian portuguese social media. In: *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. [S.l.: s.n.], 2018. p. 746–749. Citado na página 21.

COELLO, C. A. C. C.; PULIDO, G. T. A micro-genetic algorithm for multiobjective optimization. In: SPRINGER. *International conference on evolutionary multi-criterion optimization*. [S.l.], 2001. p. 126–140. Citado na página 17.

CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. *Introduction to algorithms*. [S.l.]: MIT press, 2022. Citado na página 49.

EKINCI, C.; HAKKOZ, M. A.; KIRAN, Ü.; SEKER, S. Handling missing values in mixed panel financial data: A comparison of different techniques. *PressAcademia Procedia*, PressAcademia, v. 18, n. 1, p. 103–104, 2024. Citado na página 16.

EMMANUEL, T.; MAUPONG, T.; MPOELENG, D.; SEMONG, T.; MPHAGO, B.; TABONA, O. A survey on missing data in machine learning. *Journal of Big Data*, v. 8, n. 1, p. 1–37, Dec 2021. Citado 2 vezes nas páginas 16 e 29.

ESMAEILI, A.; BEHDIN, K.; FAKHARIAN, M.; MARVASTI, F. Transductive multi-label learning from missing data using smoothed rank function. *Pattern Analysis and Applications*, p. 1225–1233, 2020. Citado na página 46.

FARHANGFAR, A.; KURGAN, L.; DY, J. Impact of imputation of missing values on classification error for discrete data. *Pattern Recognition*, v. 41, n. 12, p. 3692–705, Dec 1 2008. Citado 4 vezes nas páginas 16, 17, 24 e 33.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. From data mining to knowledge discovery in databases. 1996. Citado 2 vezes nas páginas 15 e 22.

FINATTO, M. J. B.; SILVA, A. da; ESTEVES, F. F. Fake news e desinformação sobre vacinas: contribuições dos estudos da terminologia, do texto e do discurso. *Revista GTLex*, v. 6, n. 2, p. 345, 2021. Citado na página 22.

FRANK, E.; HALL, M.; WITTEN, I. *The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques"*. Fourth. [S.l.]: Morgan Kaufmann, 2016. Citado na página 48.

FREITAS, A. A. *Data mining and knowledge discovery with evolutionary algorithms*. [S.l.]: Springer Science & Business Media, 2002. Citado na página 25.

FREITAS, S. T. d. *Análise bayesiana dos modelos de regressão linear com erros simétricos autorregressivos e dados incompletos*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2022. Citado na página 22.

FUSAR-POLI, P.; RADUA, J. Ten simple rules for conducting umbrella reviews. *Evidence-based mental health*, BMJ Publishing Group, v. 21, n. 3, p. 95, 2018. Citado na página 58.

GARCIA, J.; KALENATIC, D.; BELLO, C. Missing data imputation in multivariate data by evolutionary algorithms. *Comput Hum Behav*, p. 1468–1474, 2011. Citado na página 17.

GARCIARENA, U.; SANTANA, R. An extensive analysis of the interaction between missing data types, imputation methods, and supervised classifiers. *Expert Systems with Applications*, v. 89, p. 52–65, Dec 15 2017. Citado 5 vezes nas páginas 16, 17, 23, 29 e 33.

GHANI, M.; RAFI, M.; TAHIR, M. Discriminative adaptive sets for multi-label classification. *IEEE Access*, v. 8, p. 227579–95, Dec 1 2020. Citado na página 17.

GIBAJA, E.; VENTURA, S. A tutorial on multilabel learning. *ACM Computing Surveys (CSUR)*, v. 47, n. 3, p. 1–38, Apr 16 2015. Citado na página 28.

GONÇALVES, E.; FREITAS, A.; PLASTINO, A. A survey of genetic algorithms for multi-label classification. In: *2018 IEEE Congress on Evolutionary Computation (CEC)*. [S.l.: s.n.], 2018. p. 1–8. Citado 3 vezes nas páginas 27, 28 e 33.

GONÇALVES, E.; PLASTINO, A.; FREITAS, A. A genetic algorithm for optimizing the label ordering in multi-label classifier chains. In: *Proc. Int. Conf. Tools with Artif. Intell. ICTAI*. [S.l.: s.n.], 2013. p. 469–476. Citado 2 vezes nas páginas 33 e 44.

GONZÁLEZ, J.; ORTEGA, J.; DAMAS, M.; MARTÍN-SMITH, P. Many-objective cooperative co-evolutionary feature selection: A lexicographic approach. In: SPRINGER. *International Work-Conference on Artificial Neural Networks*. [S.l.], 2019. p. 463–474. Citado 2 vezes nas páginas 43 e 44.

GONZÁLEZ, J.; ORTEGA, J.; ESCOBAR, J.; DAMAS, M. A lexicographic cooperative co-evolutionary approach for feature selection. *Neurocomputing*, v. 463, p. 59–76, 2021. Citado na página 43.

GRZYMALA-BUSSE, J. W.; HU, M. A comparison of several approaches to missing attribute values in data mining. In: SPRINGER. *Rough Sets and Current Trends in Computing: Second International Conference, RSCTC 2000 Banff, Canada, October 16–19, 2000 Revised Papers 2*. [S.l.], 2001. p. 378–385. Citado na página 25.

HAN, J.; KAMBER, M.; PEI, J. *Data Mining: Concepts and Techniques*. 3rd. ed. [S.l.: s.n.], 2012. Citado 2 vezes nas páginas 15 e 22.

HAN, M.; WU, H.; CHEN, Z.; LI, M.; ZHANG, X. A survey of multi-label classification based on supervised and semi-supervised learning. *International Journal of Machine Learning and Cybernetics*, Springer, v. 14, n. 3, p. 697–724, 2023. Citado na página 35.

HEYMANS, M.; TWISK, J. Handling missing data in clinical research. *Journal of clinical epidemiology*, v. 151, p. 185, Nov 1 2022. Citado 2 vezes nas páginas 15 e 16.

HOLLAND, J. H. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence.* [S.l.]: U Michigan Press, 1975. Citado na página 16.

HONAKER, J.; KING, G. What to do about missing values in time-series cross-section data. *American Journal of Political Science*, v. 54, n. 2, p. 561–581, Apr 2010. Citado na página 16.

HRUSCHKA, E. R.; HRUSCHKA, E. R.; EBECKEN, N. F. Towards efficient imputation by nearest-neighbors: A clustering-based approach. In: SPRINGER. *AI 2004: Advances in Artificial Intelligence: 17th Australian Joint Conference on Artificial Intelligence, Cairns, Australia, December 4-6, 2004. Proceedings 17*. [S.l.], 2005. p. 513–525. Citado na página 25.

KAMEL, M. A. Big data analytics and market performance: the roles of customization and personalization strategies and competitive intensity. *Journal of Enterprise Information Management*, Emerald Publishing Limited, v. 36, n. 6, p. 1727–1749, 2023. Citado na página 21.

KARAFOTIAS, G.; HOOGENDOORN, M.; EIBEN, A. Evaluating reward definitions for parameter control. In: *Proceedings of the 18th European Conference on Applications of Evolutionary Computation (EvoApplications 2015)*. [S.l.]: Springer, 2015. p. 667–680. Citado na página 42.

KONG, X.; ZHOU, W.; SHEN, G.; ZHANG, W.; LIU, N.; YANG, Y. Dynamic graph convolutional recurrent imputation network for spatiotemporal traffic missing data. *Knowledge-Based Systems*, Elsevier, v. 261, p. 110188, 2023. Citado na página 15.

KRISHNASWAMY, V.; SINGH, N.; SHARMA, M.; VERMA, N.; VERMA, A. Application of crisp-dm methodology for managing human-wildlife conflicts: an empirical case study in india. *Journal of Environmental Planning and Management*, Taylor & Francis, v. 66, n. 11, p. 2247–2273, 2023. Citado na página 22.

KUMAR, M.; HUSAIN, D. M.; UPRETI, N.; GUPTA, D. Genetic algorithm: Review and application. *Available at SSRN 3529843*, 2010. Citado 2 vezes nas páginas 26 e 27.

LI, P.; STUART, E.; ALLISON, D. Multiple imputation: a flexible tool for handling missing data. *Jama*, v. 314, n. 18, p. 1966–7, Nov 10 2015. Citado na página 16.

LI, X.; LI, H.; LU, H.; JENSEN, C. S.; PANDEY, V.; MARKL, V. Missing value imputation for multi-attribute sensor data streams via message propagation. *Proc. VLDB Endow.*, VLDB Endowment, v. 17, n. 3, p. 345–358, nov 2023. ISSN 2150-8097. Disponível em: <https://doi.org/10.14778/3632093.3632100>. Citado na página 15.

LIN, W.; TSAI, C. Missing value imputation: a review and analysis of the literature (2006–2017). *Artificial Intelligence Review*, v. 53, p. 1487–509, Feb 2020. Citado 6 vezes nas páginas 15, 16, 24, 29, 34 e 58.

LING, W.; DONG-MEI, F. Estimation of missing values using a weighted k-nearest neighbors algorithm. In: IEEE. *2009 International Conference on Environmental Science and Information Application Technology*. [S.l.], 2009. v. 3, p. 660–663. Citado na página 25.

LITTLE, R. J.; RUBIN, D. B. *Statistical analysis with missing data*. [S.l.]: John Wiley & Sons, 2019. v. 793. Citado 2 vezes nas páginas 22 e 23.

LIU, W.; WANG, H.; SHEN, X.; TSANG, I. The emerging trends of multi-label learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 44, n. 11, p. 7955–74, Oct 12 2021. Citado na página 27.

LOBATO, F. *Estratégias evolucionárias para otimização no tratamento de dados ausentes por imputação múltipla de dados*. Tese (Doutorado) — Engenharia Elétrica) - Instituto de Tecnologia, Universidade Federal do Pará, 2016. Citado 12 vezes nas páginas 16, 17, 22, 23, 25, 26, 33, 37, 38, 39, 45 e 47.

LOBATO, F.; SALES, C.; ARAUJO, I.; TADAIESKY, V.; DIAS, L.; RAMOS, L.; SANTANA, A. Multi-objective genetic algorithm for missing data imputation. *Pattern Recognition Letters*, v. 68, p. 126–31, Dec 15 2015. Citado 4 vezes nas páginas 37, 39, 40 e 42.

LOBATO, F.; TADAIESKY, V.; ARAUJO, I.; SANTANA, A. d. An evolutionary missing data imputation method for pattern classification. In: *Proc. Genet Evol Comput Conf - GECCO*. [S.l.: s.n.], 2015. Citado na página 47.

LONGFORD, N. T. *Missing data and small-area estimation: Modern analytical equipment for the survey statistician*. [S.l.]: Springer Science & Business Media, 2005. Citado na página 22.

LUENGO, J.; GARCÍA, S.; HERRERA, F. On the choice of the best imputation methods for missing values considering three groups of classification methods. *Knowl Inf Syst*, v. 32, n. 1, p. 77–108, 2012. Citado 5 vezes nas páginas 16, 23, 24, 25 e 40.

MCMAHON, P.; ZHANG, T.; DWIGHT, R. Approaches to dealing with missing data in railway asset management. *IEEE Access*, v. 8, p. 48177–94, Mar 6 2020. Citado na página 16.

MIRJALILI, S. Genetic algorithm. In: *Evolutionary Algorithms and Neural Networks*. [S.l.]: Springer, 2019, (Studies in Computational Intelligence, v. 780). Citado na página 42.

MORE, K. S.; WOLKERSDORFER, C. Exploring advanced statistical data analysis techniques for interpolating missing observations and detecting anomalies in mining influenced water data. *ACS ES&T Water*, ACS Publications, 2023. Citado na página 16.

NAKAYAMA, L. F.; RESTREPO, D.; MATOS, J.; RIBEIRO, L. Z.; MALERBI, F. K.; CELI, L. A.; REGATIERI, C. S. Brset: A brazilian multilabel ophthalmological dataset of retina fundus photos. *medRxiv*, Cold Spring Harbor Laboratory Press, p. 2024–01, 2024. Citado na página 35.

NGUYEN, T.; LUONG, A.; NGUYEN, Q.; LIEW, A.; STANTIC, B. Multi-label classification via label correlation and first order feature dependence in a data stream. *Pattern Recognition*, v. 90, p. 35–51, Jun 1 2019. Citado 3 vezes nas páginas 27, 28 e 29.

NUGROHO, H.; SURENDRO, K. A comprehensive bibliometric analysis of missing value imputation. *IEEE Access*, IEEE, 2024. Citado 2 vezes nas páginas 34 e 58.

NUNES, L.; KLUCK, M.; FACHEL, J. Use of multiple imputation for missing data: a simulation using epidemiological data. *Cad Saúde Pública [online]*, v. 25, n. 2, p. 268–278, 2009. Citado 2 vezes nas páginas 16 e 25.

PEREIRA, R.; PLASTINO, A.; ZADROZNY, B.; MERSCHMANN, L. Correlation analysis of performance measures for multi-label classification. *Information Processing & Management*, v. 54, n. 3, p. 359–69, May 1 2018. Citado na página 28.

PRESS, G. *Cleaning Big Data: Most Time-Consuming, Least Enjoyable Data Science Task, Survey Says*. 2016. [Accessed: 15-Jun-2021]. Disponível em: <https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/>. Citado na página 15.

PROVOST, F.; FAWCETT, T. Data science and its relationship to big data and data-driven decision making. *Big Data*, v. 1, n. 1, p. 51–59, 2013. Citado 2 vezes nas páginas 15 e 22.

PROVOST, F.; SAAR-TSECHANSKI, M. Handling missing values when applying classification models. *Journal of Machine Learning Research*, n. 8, 2007. Citado 2 vezes nas páginas 17 e 33.

QIAN, K.; MIN, X.; CHENG, Y.; SONG, G.; MIN, F. Self-dependence multi-label learning with double k for missing labels. *Artificial Intelligence Review*, p. 1–38, Oct 23 2022. Citado na página 28.

QIAN, K.; MIN, X.-Y.; CHENG, Y.; SONG, G.; MIN, F. Self-dependence multi-label learning with double k for missing labels. *Artificial Intelligence Review*, Springer, v. 56, n. 6, p. 5057–5094, 2023. Citado na página 36.

READ, J.; PFAHRINGER, B.; HOLMES, G.; FRANK, E. Classifier chains for multi-label classification. In: SPRINGER. *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2009, Bled, Slovenia, September 7-11, 2009, Proceedings, Part II 20*. [S.l.], 2009. p. 254–269. Citado na página 32.

READ, J.; PFAHRINGER, B.; HOLMES, G.; FRANK, E. Classifier chains for multi-label classification. *Machine Learning*, v. 85, p. 333–59, Dec 2011. Citado 5 vezes nas páginas 17, 29, 32, 35 e 47.

REN, L.; WANG, T.; SEKLOULI, A.; ZHANG, H.; BOURAS, A. A review on missing values for main challenges and methods. *Information Systems*, Oct 2023. Citado 3 vezes nas páginas 16, 24 e 58.

REY-BLANCO, D.; ARBUÉS, P.; LÓPEZ, F. A.; PÁEZ, A. Using machine learning to identify spatial market segments. a reproducible study of major spanish markets. *Environment and Planning B: Urban Analytics and City Science*, SAGE Publications Sage UK: London, England, v. 51, n. 1, p. 89–108, 2024. Citado na página 15.

REYNOSO-MEZA, G.; SANCHIS, J.; BLASCO, X.; HERRERO, J. M. Hybrid de algorithm with adaptive crossover operator for solving real-world numerical optimization problems. In: *2011 IEEE Congress of Evolutionary Computation (CEC)*. [S.l.]: IEEE, 2011. p. 1551–1556. Citado na página 42.

REZENDE, S. O. *Intelligent systems: concepts and applications*. 1st. ed. [S.l.]: Manole, 2003. Citado 3 vezes nas páginas 15, 22 e 26.

RIEMENSCHNEIDER, M.; HERBST, A.; RASCH, A.; GORLATCH, S.; HEIDER, D. ecccl: parallelized gpu implementation of ensemble classifier chains. *BMC bioinformatics*, Springer, v. 18, p. 1–4, 2017. Citado 2 vezes nas páginas 9 e 32.

RUBIN, D. An overview of multiple imputation. In: *Proceedings of the survey research methods section of the American statistical association*. Princeton, NJ, USA: [s.n.], 1988. v. 79, p. 84. Citado na página 16.

RUBIN, D. *Multiple imputation for nonresponse in surveys*. [S.l.]: John Wiley & Sons, 2004. Citado 2 vezes nas páginas 16 e 25.

SÁ, A. de; PIMENTA, C.; PAPPA, G.; FREITAS, A. A robust experimental evaluation of automated multi-label classification methods. In: *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*. [S.l.: s.n.], 2020. p. 175–183. Citado 2 vezes nas páginas 27 e 28.

SA'ADI, Z.; YASEEN, Z. M.; FAROOQUE, A. A.; MOHAMAD, N. A.; MUHAMMAD, M. K. I.; IQBAL, Z. Long-term trend analysis of extreme climate in sarawak tropical peatland under the influence of climate change. *Weather and Climate Extremes*, Elsevier, v. 40, p. 100554, 2023. Citado na página 15.

SAINANI, K. L. Dealing with missing data. *PM&R*, Elsevier, v. 7, n. 9, p. 990–994, 2015. Citado na página 23.

SANTOS, M.; PEREIRA, R.; COSTA, A.; SOARES, J.; SANTOS, J.; ABREU, P. Generating synthetic missing data: A review by missing mechanism. *IEEE Access*, v. 7, p. 11651–11667, 2019. Citado 2 vezes nas páginas 16 e 47.

SCHMITT, P.; MANDEL, J.; GUEDJ, M. A comparison of six methods for missing data imputation. *Journal of Biometrics & Biostatistics*, v. 6, n. 1, p. 1, 2015. Citado na página 55.

SCHOUTEN, R. M.; LUGTIG, P.; VINK, G. Generating missing values for simulation purposes: a multivariate amputation procedure. *Journal of Statistical Computation and Simulation*, Taylor & Francis, v. 88, n. 15, p. 2909–2930, 2018. Citado na página 24.

SEMENKIN, E.; SEMENKINA, M. Self-configuring genetic algorithm with modified uniform crossover operator. In: *Advances in Swarm Intelligence. ICSI 2012. Lecture Notes in Computer Science*. [S.l.]: Springer, 2012. v. 7331, p. 414–421. Citado na página 42.

SESTINO, A.; PRETE, M. I.; PIPER, L.; GUIDO, G. Internet of things and big data as enablers for business digitalization strategies. *Technovation*, Elsevier, v. 98, p. 102173, 2020. Citado na página 15.

SETTOUTI, N.; DOUIBI, K.; BECHAR, M. E. A.; DAHO, M. E. H.; SAIDI, M. Semi-supervised learning with collaborative bagged multi-label k-nearest-neighbors. *Open Computer Science*, De Gruyter Open, v. 9, n. 1, p. 226–242, 2019. Citado 2 vezes nas páginas 9 e 31.

SEVILLA, J.; HEIM, L.; HO, A.; BESIROGLU, T.; HOBBHAHN, M.; VILLALOBOS, P. Compute trends across three eras of machine learning. In: IEEE. *2022 International Joint Conference on Neural Networks (IJCNN)*. [S.l.], 2022. p. 1–8. Citado na página 21.

SHAHZAD, W.; REHMAN, Q.; AHMED, E. Missing data imputation using genetic algorithm for supervised learning. *Int J Adv Comput Sci Appl*, v. 8, 2017. Citado 2 vezes nas páginas 37 e 39.

SHARMA, A.; RANI, S.; SAH, D. K.; KHAN, Z.; BOULILA, W. Homlc-hyperparameter optimization for multi-label classification of intrusion detection data for internet of things network. *Sensors*, MDPI, v. 23, n. 19, p. 8333, 2023. Citado na página 35.

SHU, X.; YE, Y. Knowledge discovery: Methods from data mining and machine learning. *Social Science Research*, Elsevier, v. 110, p. 102817, 2023. Citado 2 vezes nas páginas 15 e 22.

SINGH, R.; SINGH, R. Applications of sentiment analysis and machine learning techniques in disease outbreak prediction – a review. *Materials Today: Proceedings*, v. 81, p. 1006–1011, 2023. ISSN 2214-7853. International Virtual Conference on Sustainable Materials (IVCSM-2k20). Disponível em: <https://www.sciencedirect.com/science/article/pii/S2214785321032764>. Citado na página 15.

SMITH, D. R. The design of divide and conquer algorithms. *Science of Computer Programming*, Elsevier, v. 5, p. 37–58, 1985. Citado na página 29.

SONG, R.; LIU, Z.; CHEN, X.; AN, H.; ZHANG, Z.; WANG, X.; XU, H. Label prompt for multi-label text classification. *Applied Intelligence*, Springer, v. 53, n. 8, p. 8761–8775, 2023. Citado na página 35.

SOUSA, G. N. de; GUIMARÃES, I. da S.; VIANA, J. A. N.; REINHOLD, O.; JUNIOR, A. F. L.; LOBATO, F. M. F. Análise do setor de telecomunicação brasileiro: Uma visão sobre reclamações. *Revista Ibérica de Sistemas e Tecnologias de Informação*, Associação

Ibérica de Sistemas e Tecnologias de Informacao, n. 37, p. 31–48, 2020. Citado na página 22.

SUN, L.; YIN, T.; DING, W.; QIAN, Y.; XU, J. Feature selection with missing labels using multilabel fuzzy neighborhood rough sets and maximum relevance minimum redundancy. *IEEE Transactions on Fuzzy Systems*, v. 30, n. 5, p. 1197–211, Jan 22 2021. Citado na página 28.

SZCZEPAŃSKI, M.; PAWLICKI, M.; KOZIK, R.; CHORAŚ, M. The application of deep learning imputation and other advanced methods for handling missing values in network intrusion detection. *Vietnam Journal of Computer Science*, World Scientific, v. 10, n. 01, p. 1–23, 2023. Citado na página 16.

TANG, L.; RAJAN, S.; NARAYANAN, V. K. Large scale multi-label classification via metalabeler. In: *Proceedings of the 18th international conference on World wide web*. [S.l.: s.n.], 2009. p. 211–220. Citado 2 vezes nas páginas 27 e 29.

TCHUENTE, D.; HADDADI, A. E. One decade of big data for firms' competitiveness: insights and a conceptual model from bibliometrics. *Journal of Enterprise Information Management*, Emerald Publishing Limited, v. 36, n. 6, p. 1421–1453, 2023. Citado 2 vezes nas páginas 15 e 21.

TRAN, C.; ZHANG, M.; ANDREAE, P. Multiple imputation for missing data using genetic programming. In: *Proceedings of the 2015 annual conference on genetic and evolutionary computation*. [S.l.: s.n.], 2015. p. 583–590. Citado 2 vezes nas páginas 36 e 39.

TRIGUERO, I.; GONZáLEZ, S.; MOYANO, J.; GARCÍA, S.; ALCALá-FDEZ, J.; LUENGO, J.; AL. et. Keel 3.0: An open source software for multi-stage analysis in data mining. *Int J Comput Intell Syst*, v. 10, p. 1238–1249, 2017. Citado na página 48.

TSAI, C.; LI, M.; LIN, W. A class center based approach for missing value imputation. *Knowledge-Based Systems*, v. 151, p. 124–35, Jul 2018. Citado 2 vezes nas páginas 15 e 16.

TSOUMAKAS, G.; KATAKIS, I. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, IGI Global, v. 3, n. 3, p. 1–13, 2007. Citado na página 29.

TSOUMAKAS, G.; KATAKIS, I.; VLAHAVAS, I. Effective and efficient multilabel classification in domains with a large number of labels. In: *Proc. ECML/PKDD 2008 Work. Min. Multidimens. Data*. [S.l.: s.n.], 2008. p. 30–44. Citado 6 vezes nas páginas 9, 29, 30, 31, 33 e 47.

TSOUMAKAS, G.; KATAKIS, I.; VLAHAVAS, I. Random k-labelsets for multilabel classification. *IEEE Transactions on Knowledge and Data Engineering*, v. 23, n. 7, p. 1079–89, Sep 9 2010. Citado 2 vezes nas páginas 27 e 29.

TSOUMAKAS, G.; SPYROMITROS-XIOUFIS, E.; VILCEK, J.; VLAHAVAS, I. Mulan: A java library for multi-label learning. *Journal of Machine Learning Research*, v. 12, p. 2411–2414, 2011. Citado na página 48.

VANEGAS, C. E. D.; MEJÍA, J. C. G.; AGUDELO, F. A. V.; DURAN, D. E. S. A representation based on essence for the crisp-dm methodology. *Computación y Sistemas*, Instituto Politécnico Nacional, Centro de Investigación en Computación, v. 27, n. 3, p. 675–689, 2023. Citado na página 22.

VENKATESAN, R.; ER, M. Multi-label classification method based on extreme learning machines. In: *2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*. [S.l.: s.n.], 2014. p. 619–624. Citado na página 27.

VIDULIN, V. Searching for credible relations in machine learning. *Informatica*, v. 37, n. 3, 2013. Citado 2 vezes nas páginas 9 e 30.

WANG, C.; LIN, Y.; LIU, J. Feature selection for multi-label learning with missing labels. *Applied Intelligence*, v. 49, p. 3027–42, Aug 15 2019. Citado 3 vezes nas páginas 35, 36 e 46.

WIRTH, R.; HIPP, J. Crisp-dm: Towards a standard process model for data mining. In: MANCHESTER. *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*. [S.l.], 2000. v. 1, p. 29–39. Citado na página 22.

WU, X.; ZHU, X.; WU, G.-Q.; DING, W. Data mining with big data. *IEEE transactions on knowledge and data engineering*, IEEE, v. 26, n. 1, p. 97–107, 2013. Citado na página 21.

ZHANG, M.-L.; ZHOU, Z.-H. Ml-knn: A lazy learning approach to multi-label learning. *Pattern recognition*, Elsevier, v. 40, n. 7, p. 2038–2048, 2007. Citado na página 31.

ZHANG, N.; LIU, C.; STEINER, S. J.; COLLETTI, R. B.; BALDASSANO, R.; CHEN, S.; COHEN, S.; KAPPELMAN, M. D.; SAEED, S.; CONKLIN, L. S. et al. Using multiple imputation of real-world data to estimate clinical remission in pediatric inflammatory bowel disease. *Journal of Comparative Effectiveness Research*, Becaris Publishing Ltd Royston, UK, v. 12, n. 4, p. e220136, 2023. Citado na página 15.

ZHENG, X.; LI, P.; CHU, Z.; HU, X. A survey on multi-label data stream classification. *IEEE Access*, v. 8, p. 1249–75, Dec 24 2019. Citado na página 29.

ZUO, J.; ZEITOUNI, K.; TAHER, Y.; GARCIA-RODRIGUEZ, S. Graph convolutional networks for traffic forecasting with missing values. *Data Mining and Knowledge Discovery*, Springer, v. 37, n. 2, p. 913–947, 2023. Citado na página 15.

APÊNDICE A - Teste com taxa de mutação

A taxa de mutação é mais elevada do que as taxas de uso comuns, pois o ponto de partida não é aleatório. Nesse sentido, a Figura 6 apresenta os testes com diferentes taxas de mutação e a performance da acurácia do modelo. Para esse teste, Um dos conjuntos de dados utilizados no artigo ("flags") foi escolhido, e a taxa de mutação variou de 0, 5, 10, 15, 20, 25, 30, 35 a 40%.

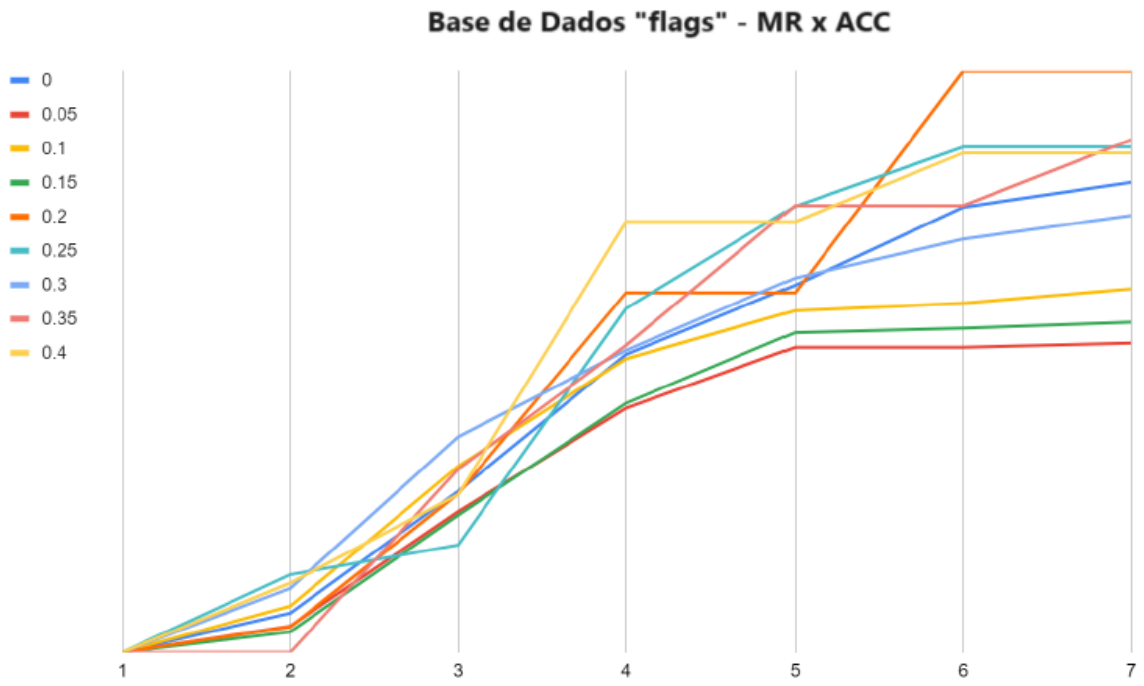


Figura 6 – Testes com diferentes taxas de mutação.

Como pode ser observado no gráfico, os melhores resultados foram obtidos com uma taxa de mutação de 20%. Nesse sentido, considerando que a população inicial é obtida por outros métodos, os experimentos de parametrização demonstraram que uma MR mais alta proporciona melhores resultados, garantindo uma rápida convergência.

APÊNDICE B - Fluxograma do EvoImp

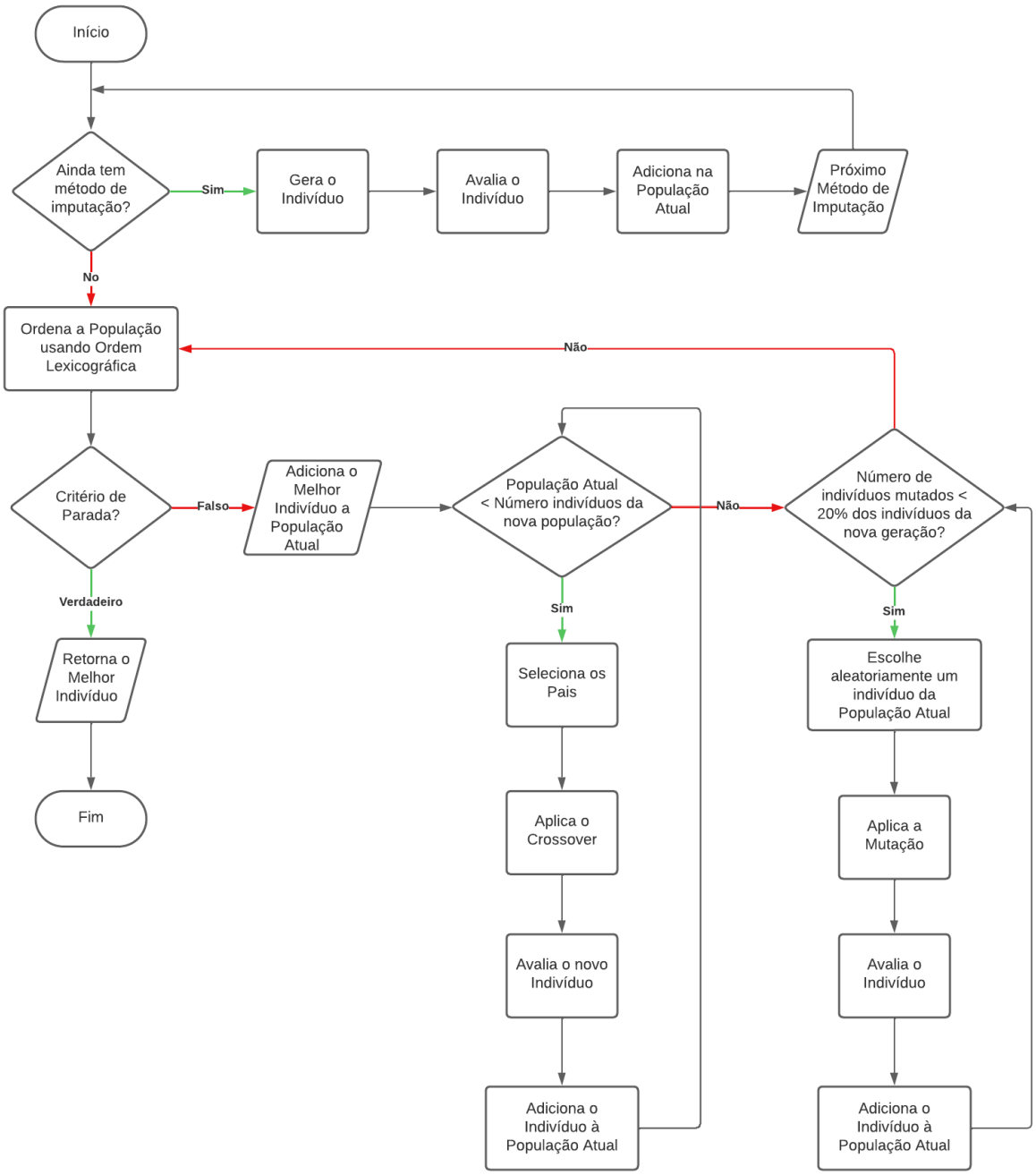


Figura 7 – Fluxograma de execução do EvoImp

APÊNDICE C - *Baseline*

Tabela 12 – Testes do EvoImp e dos classificadores com os datasets sem dados ausentes (*baseline*)

	Exact Match						Accuracy						Hamming Loss					
Db	BR	HOMER	ML-KNN	CC	ECC	EVOIMP	BR	HOMER	ML-KNN	CC	ECC	EVOIMP	BR	HOMER	ML-KNN	CC	ECC	EVOIMP
b	51.95%	50.45%	50.88%	51.45%	**55.36%**	54.69%	63.62%	61.44%	61.11%	62.91%	**67.18%**	65.32%	4.94%	5.78%	4.72%	4.99%	4.19%	**3.65%**
c	33.90%	36.88%	36.08%	34.97%	35.40%	**42.71%**	44.51%	34.97%	60.20%	40.24%	53.97%	**73.65%**	16.15%	21.04%	13.88%	17.60%	14.51%	**9.78%**
e	51.22%	51.57%	56.95%	50.48%	56.60%	**59.32%**	58.08%	56.76%	68.83%	57.38%	64.35%	**69.43%**	24.74%	25.25%	19.51%	25.50%	20.21%	**18.22%**
f	60.87%	61.05%	63.52%	61.07%	64.62%	**67.09%**	70.89%	68.36%	73.01%	69.56%	72.58%	**77.47%**	26.27%	26.24%	25.27%	27.00%	24.11%	**22.24%**
s	56.80%	54.57%	66.49%	58.13%	64.01%	**72.58%**	55.28%	51.97%	69.34%	60.78%	64.85%	**79.04%**	13.68%	14.48%	8.62%	14.44%	9.37%	**6.12%**
y	50.44%	51.49%	56.96%	49.78%	54.59%	**63.09%**	61.15%	58.54%	71.98%	56.33%	68.50%	**76.12%**	24.54%	26.19%	19.33%	26.82%	20.70%	**15.14%**

ANEXO I - Artigo “EvoImp: Multiple Imputation of Multi-label Classification data with a genetic algorithm”

PLOS ONE

RESEARCH ARTICLE

EvoImp: Multiple Imputation of Multi-label Classification data with a genetic algorithm

Antonio Fernando Lavareda Jacob Junior[1,2], **Fabricio Almeida do Carmo**[2], **Adamo Lima de Santana**[3], **Ewaldo Eder Carvalho Santana**[1,2], **Fabio Manoel Franca Lobato**[2,4]*

1 Graduate Program in Electrical Engineering (PPGEE), Federal University of Maranhão (UFMA), São Luís, Maranhão, Brazil, **2** Graduate Program in Computer Engineering and Systems (PECS), State University of Maranhão (UEMA), São Luís, Maranhão, Brazil, **3** Corporate ReD Headquarters Fuji Electric Co., Tokyo, Japan, **4** Institute of Engineering and Geosciences, Federal University of Western Pará (UFOPA), Santarém, Pará, Brazil

* fabio.lobato@ufopa.edu.br

Citation: Jacob Junior AFL, do Carmo FA, de Santana AL, Santana EEC, Lobato FMF (2024) EvoImp: Multiple Imputation of Multi-label Classification data with a genetic algorithm. PLoS ONE 19(1): e0297147. https://doi.org/10.1371/journal.pone.0297147

Editor: Mohammad A. Al-Mamun, West Virginia University, UNITED STATES

Received: May 29, 2023

Accepted: December 28, 2023

Published: January 19, 2024

Peer Review History: PLOS recognizes the benefits of transparency in the peer review process; therefore, we enable the publication of all of the content of peer review and author responses alongside final, published articles. The editorial history of this article is available here: https://doi.org/10.1371/journal.pone.0297147

Data Availability Statement: All relevant files are available from the Zenodo database (Link: https://doi.org/10.5281/zenodo.7748933).

Abstract

Missing data is a prevalent problem that requires attention, as most data analysis techniques are unable to handle it. This is particularly critical in Multi-Label Classification (MLC), where only a few studies have investigated missing data in this application domain. MLC differs from Single-Label Classification (SLC) by allowing an instance to be associated with multiple classes. Movie classification is a didactic example since it can be "drama" and "bibliography" simultaneously. One of the most usual missing data treatment methods is data imputation, which seeks plausible values to fill in the missing ones. In this scenario, we propose a novel imputation method based on a multi-objective genetic algorithm for optimizing multiple data imputations called Multiple Imputation of Multi-label Classification data with a genetic algorithm, or simply EvoImp. We applied the proposed method in multi-label learning and evaluated its performance using six synthetic databases, considering various missing values distribution scenarios. The method was compared with other state-of-the-art imputation strategies, such as K-Means Imputation (KMI) and weighted K-Nearest Neighbors Imputation (WKNNI). The results proved that the proposed method outperformed the baseline in all the scenarios by achieving the best evaluation measures considering the Exact Match, Accuracy, and Hamming Loss. The superior results were constant in different dataset domains and sizes, demonstrating the EvoImp robustness. Thus, EvoImp represents a feasible solution to missing data treatment for multi-label learning.

Introduction

Missing data is ubiquitous in data analysis [1]. Their causes are the most diverse and related to the application domain. These include drawbacks in data acquisition, measurement errors, sensor network problems, data migration failures, and unwillingness to respond to survey questions [2, 3]. Since data analysis algorithms/methods are not designed to deal with Missing Values (MVs), it is essential to treat them before aiming to guarantee the results' validity, impairing the research conclusions [1, 4, 5]. MVs are problematic because of the risk of bias, which depends on the type of missing data, the extent of the missingness, and how to deal with

Funding: FMFL was financed in part by the National Council for Scientific and Technological Development (CNPq, Brazil) under Grant 147336/2020-1. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

MVs in the analyses [1]. Thus, it is critical to deal with the missing data timely for intelligent decision-making [6].

Several techniques have emerged to address this problem [4, 7, 8]. LIN [4] comments that if the MVs rate is less than 10% or 15%, they can be removed without causing any significant loss to the mining process. However, this does not mean that the datasets in any problem domain must follow this rule; in other words, small amounts of missing data may contain essential information that must be managed [9]. In addressing this issue, the literature suggests using missing data imputation methods, which involve replacing missing data with actual (plausible) values. While this approach allows for more data retention compared to deletion, it requires time to generate reasonable replacement values [10, 11].

A naïve method for tackling the missing values issue is by Single Imputation (SI). This method involves filling in missing values with a single estimated value, often based on mean, median, or regression models [4]. While this approach simplifies the dataset and makes it easier to analyze, it can introduce bias and underestimate uncertainty in the results [12, 13]. To overcome this limitation, Rubin [14] introduced a gold-standard imputation strategy within the scientific community—Multiple Imputations (MI) for handling missing data. In contrast with SI approaches, this method seeks to find a single solution in which m complete solutions are created in the operational database such that $m > 1$. These solutions were analyzed separately and combined to obtain the best solution [15, 16]. To reduce the missing values prediction error, using metaheuristics could optimize the value that would be imputed [15]. Notably, bioinspired strategies such as Genetic Algorithms (GAs) are prominent in optimizing solutions [17].

The GAs were proposed by Holland [18]. It is an optimization heuristic based on "the survival of the fittest", inspired in Charles Darwin´s evolutionary theory. Regarding the GAs usage for Multiple Imputations, it is crucial to acknowledge the work of Garcia [19] and the MultImp algorithm [15]. The MultImp algorithm serves as the cornerstone for this research. This algorithm employed genetic algorithms for multiple imputations and was also applied for Multi-Label Classification (MLC) scenarios. The authors contend that data mining tasks, particularly those related to data classification, are notably sensitive to addressing MV. Furthermore, classification tasks are widely used to assess the accuracy (ACC) of imputation methods [5, 11, 20]. Consequently, the higher the classification accuracy, the more successful the imputation method. However, only a few studies have employed MLC. In contrast to Single-Label Classification (SLC), or simply data classification, which associates an example with a single label, MLC allows an instance to be associated with multiple labels, thereby increasing the complexity of classification tasks [21, 22]. Further details on this topic will be highlighted in the Background section.

Considering the importance of handling missing values in data analysis and the available solutions in the existing literature, this work presents an efficient algorithmic approach for multiple imputations applied to multi-label classification tasks. This method is named EvoImp, a combination of "*evolutionary*" and "*imputation*". Furthermore, the name is inspired by MultImp [15], which serves as the foundation for our algorithm and has shown promise in its preliminary stages for multiple imputations with missing data. EvoImp enhances the parameterization of MultImp to maximize its imputation capabilities and explores new configurations for computational experiments.

We conducted a rigorous benchmarking process to validate the proposed method's performance using diverse multi-label datasets. We compared EvoImp with well-established imputation methods documented in the literature. These datasets were systematically subjected to six missing value rates to simulate the Missing Completely At Random (MCAR) mechanism. The outcomes of these experiments were meticulously evaluated using five distinct classifiers. This

comprehensive evaluation provides insights into the strengths and potential limitations of our EvoImp when applied to real-world multi-label classification scenarios. By addressing the challenges associated with missing data in this context, our work aims to advance multi-label classification and the broader field of data analysis.

Accordingly, the remainder of this paper is organized as follows. The section "Background" presents a preliminary background. The section "EvoImp—Proposed Method" included the proposed method in this section. The section "Computational Experiments" details the experimental setup. The performance of the method and comparison with data imputation techniques are demonstrated in sections "Results and Analysis" and "Discussion". Finally, section "Conclusion and Suggestions for Future Work" summarizes the paper and points out potential directions for future exploration.

Background

Multi-label Classification and classical approaches to handling MVs

In single-label classification problems, a set of class labels is predetermined, and each object must be associated with one and only one label [23]. Formally, let X denote the input/feature space, and y denote the class value, where $y \in L$, which is the output space (a set of disjoint class labels). In this case, each sample is strictly associated with a single class label [24, 25]

However, there are increasingly more contexts in which data may belong to more than one class label. This classification condition is referred to as Multi-Label classification. Initially, MLC primarily focused on tasks such as text categorization, protein function classification, music categorization, semantic scene classification, and medical diagnosis [23, 24, 26]. Recently, new applications have emerged in Computer Vision, Natural Language Processing, and Data Mining, including Video Annotation, Legal Text Mining, and User Profiling [27]. According to [25, 28], similar to SLC, MLC is represented by X and y, where each sample $x \in X$ is assigned a subset of the output space (a set of non-disjoint class labels). Table 1 illustrates a toy example depicting the difference between SLC and MLC, adapted from [29]. Considering that the data in Table 1 comprises 5 instances (x_1, x_2, x_3, x_4, x_5) and 3 labels (y_1, y_2, y_3).

Table 1a illustrates the SLC scenario, where five data instances (x_1 to x_5) are each strictly associated with a single label (y_1 to y_3). For instance, x_1 is associated with y_1, x_2 is associated with y_2, and so on. On the other hand, MLC allows data instances to be associated with

Table 1. Comparison of SLC and MLC using a toy example with 5 instances and 3 labels.

Data	Label
x_1	y_1
x_2	y_2
x_3	y_3
x_4	y_1
x_5	y_3

(a) Single-label

Data	Labels
x_1	y_1, y_2
x_2	y_2, y_3
x_3	y_1, y_3
x_4	y_2
x_5	y_3

(b) Multi-label

https://doi.org/10.1371/journal.pone.0297147.t001

multiple labels simultaneously. Table 1b demonstrates the MLC scenario, where the same five data instances (x_1 to x_5) can have multiple labels assigned to them. For example, x_1 is associated with both y_1 and y_2, x_2 is associated with both y_2 and y_3, and so forth. This distinction highlights how SLC restricts each data instance to a single label, while MLC permits instances to belong to multiple labels simultaneously, making it more suitable for scenarios where objects or data points can be associated with different classes.

Although the difference is subtle in theory, MLC tends to be more challenging in practice. Gonçalves et al. [23] and Sá et al. [25] enumerated the following reasons for this:

- The possible classes of a given instance (output space) in MLC grow exponentially from the increasing number of labels. Therefore, when considering that a problem has L distinct labels, the size of the output space in MLC is 2^L (combination of labels) while it is only L in SLC;
- An MLC algorithm must consider whether there exists or not a correlation between labels. This kind of correlation is an essential step to ensure the effectiveness of several MLC processes [24, 30, 31];
- MLC systems performance evaluation uses different metrics than those traditionally used in SLC [32]. In SLC, the rating of a new instance can be either correct or wrong. On the other hand, in MLC, the result can be partially correct. It occurs when the classifier predicts some correct labels but includes some incorrect predictions or even omits a label that should be predicted. This problem requires cautious attention since some metrics follow contrasting aspects to define what is a good MLC prediction [25, 33];
- Unlike SLC problems, which traditionally involve the analysis of relational (structured) data, MLC applications typically address big data tasks, which involve semi-structured or unstructured data [24, 34].

All these challenges have amplified the complexity associated with handling MVs. Nevertheless, finding studies that relate MLC and MV is not straightforward, as demonstrated in [4, 8, 17].

In this context, we emphasize a limited number of studies that specifically address the issue of missing labels [35, 36], which means focusing on predicting an unknown label. Wang et al. [35] present a multi-label feature selection that considers feature interaction. For that, the authors use the definitions of multi-label neighborhood information entropy and multi-label neighborhood mutual information to mitigate the negative impact of missing labels. Cheng, Song & Qian [36] focus on addressing missing labels by leveraging label correlations and implementing a two-level kernel extreme learning machine autoencoder. The authors verified the proposed method on both missing and complete label datasets. Since these studies primarily focus on missing labels rather than missing values (predictive features), to the best of our knowledge, there is no work addressing missing values in the predictive feature space in an ML scenario. Thus, this constitutes one of the contributions of the present study.

Bio-inspired computation for the handling of MVs

Tran, Zang, and Andreae [37] proposed a data imputation method by adopting an approach based on genetic programming called GPMI. An MI strategy was applied in this method, and an estimation of missing values was performed using regression techniques. The GPMI was compared with seven imputation methods through an experiment carried out in eight datasets and applying seven different missing values ratios (5, 10, 20, 30, 40, and 50) with the aid of

MCAR as a missing data mechanism. The classifier's accuracy was the performance measure adopted. The results suggest that the planned method performed better than all methods. According to the authors, genetic programming was primarily responsible for these results because the algorithm initially used random samples to fill the gaps before being submitted to genetic processes. The results confirmed that strategies based on evolutionary algorithms are feasible alternatives for missing values treatment.

Shahzad, Rehman, and Ahmed, in their study, "Missing Data Imputation using Genetic Algorithm for Supervised Learning" [38], employed GA to search for plausible values for missing data imputation. An exciting strategy adopted in this study is using information gain to observe how solutions are found as the process grows. In an experiment with five datasets that originally contained missing values, the proposed method was compared with other imputation approaches: the average, lowest value, highest value, zero, and MI. They used the following performance measures: predictive accuracy, precision, recall, F-measure, and the area under the Receiver Operating Characteristic (ROC) curve, with the following classifiers: NB-tree, PART, JRIP, Naive Bayes, KNN, and J48. The authors noted that the GA-based method showed promising results and worked well in datasets with a high percentage of missing values.

In [39], an algorithm called MOGAImp was proposed for multiple imputation datasets based on genetic algorithms. One of the exciting strategies of this work is to apply a multi-objective approach, which until then had not been adopted in the literature for the performance analysis of imputation techniques. This approach involves simultaneously employing two or more evaluation measures. It can be explained by the fact that there are distinctions between various performance measures because, while one increases, the other declines. In the case of MOGAImp, two conflicting measures were used: the classifier accuracy and the predictive accuracy of the imputation method, calculated using Normalized Root-Mean-Square error (NRMSE) and the Pareto front.

Another critical factor in the study conducted by [39] concerns population initialization, which employs a pool of candidate solutions based on each attribute. The solution pool involves grouping all possible dataset values for the attribute that has a missing value (by lexicographically comparing two strings in cases of categorical variables). The method was experimentally compared with other well-known techniques in the literature, employing benchmarking through several databases with missing values. The results demonstrated that the method achieved competitive performance and, according to the authors, demonstrated potential for real-world applications. However, high computational power is required for handling the MVs individually with MOGAImp and through the solution pool. Additionally, this strategy is an excellent alternative to a mixture of genetic materials. Therefore, it has been adopted in EvoImp as a baseline for mutation operations.

In [15], the authors created a scheme based on genetic algorithms, which served as a baseline for developing and analyzing the method employed in this study. The strategy, nominated as MultImp, predicts multiple imputations of datasets in a multi-label classification model. In this study, the authors conducted experiments using four databases that were initially completed. Subsequently, 5% of the missing values were added through the MCAR mechanism. Binary relevance (BR) was employed as the multi-label classifier, with C4.5 as a parameter. In the test scenario, MultImp was compared with two other imputation methods (K-Nearest Neighbors Imputation—KNNI and Most Common—MC) and evaluated lexicographically using the following measures: Exact Match (EM), Accuracy, and Hamming Loss (HL). The preliminary results of this study proved to be promising, particularly in the case of EM, where the performance achieved by the method was better in all the datasets used and justified adopting the lexicographical approach.

For a comprehensive summary of the works discussed in this section, we have provided a detailed table in our supplementary material, available on the project's GitHub repository (https://github.com/jacobjr/EvoImp).

EvoImp—Proposed method

Since EvoImp is based on a genetic algorithm, the following descriptions explain how EvoImp was mapped and configured within the GA structure: a) the codification of individuals, b) the formation of the initial population, c) the configuration of genetic operators, and d) the definition of the fitness function. Fig 1 presents a toy example of this structure, which will be detailed in the following subsections.

Individual encoding and population initialization

The individual encoding of EvoImp took place in the following form: the variables in the datasets represent individual genes. Genes initially marked with "?" represent the missing values (Fig 1(a)). Each individual is represented by a completed ("accomplished") instance of the databases (Fig 1(b)). The phenotype consists of imputed values, while the genotype represents these values in binary form, as illustrated in Fig 1(c).

The initial population comprised five simple imputation methods for the generation of each individual (Fig 1(d)). All imputation methods are well-known and established in the literature [7]: K-means Clustering Imputation, KNNI, WKNNI, Concept Most Common (CMC), and MC. The parameters for the KNNI, WKNNI, and KMI methods followed the guidelines set by the authors. This kind of population initialization was adopted in EvoImp to reduce the search space and, hence, the computational costs.

The methods employed are as follows [7]:

- **KNNI:** Whenever there is a missing value, the K-nearest neighbors closest to the instance containing the MV are determined. The most common value among the K-nearest neighbors was used to impute nominal attributes. For numerical attributes, imputation is performed by calculating the average of the neighboring values;
- **WKNNI:** This technique involves determining the distances between K-nearest neighbors and a weighting distribution regarding the distances between each neighbor. After this, the KNNI process was repeated;
- **KMI:** This technique divides a database into clusters based on their features. Once this has been done, the K-nearest neighbors technique is applied when deciding which value should be imputed;
- **MC:** In this method, the most common value is adopted for imputation in nominal attributes and the average of all corresponding attributes in the case of numerical attributes;
- **CMC:** This method does the same thing as MC but only employs the referenced attribute class with MV.

In contrast to MOGAImp [39], which employs random initialization of the initial population, the proposed method optimizes simple imputations through evolutionary processes to perform multiple imputations. This approach reduces the search space and introduces a novel method. This reduction in search space is particularly beneficial in scenarios where computational cost is critical in objective function calculations, such as multi-label classification.

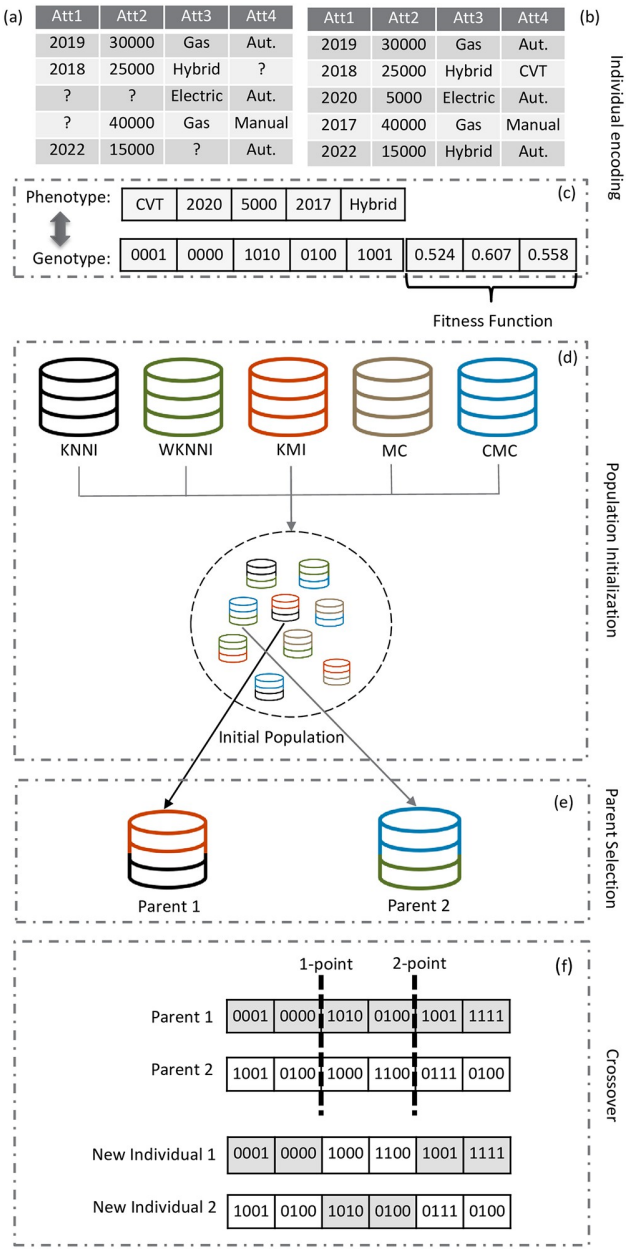


Fig 1. EvoImp's GA structure example. Toy example of a dataset with MV and how EvoImp's GA works with it. (a) Dataset with missing values; (b) A complete dataset with imputed data. (c) Phenotype: contains the values corresponding to the missing data space; Genotype: represents the genes in binary code and the values of the measurements used in the fitness function. (d) Illustration of how the initial population is initialized. (e) Random selection of parents for crossover. (f) Illustration of crossover being applied to the two selected individuals.

https://doi.org/10.1371/journal.pone.0297147.g001

It is also noteworthy that the presented work has two innovative contents: 1) using simple imputation methods as *a priori* solution, reducing the search space; 2) treating missing values in the multi-label scenario. To our best knowledge, there is no similar study in the literature.

Genetic operators

The individual selection involves a tournament in which two (or more) members of the previous population are selected, and the better one is chosen based on fitness value, as illustrated in Fig 1(e). This procedure was followed until a limited number of individuals from the current generation were obtained. The best individual is always selected through elitism [40].

In the literature, numerous methods for parameter tuning and control have been proposed and analyzed. [41] describes some of these methods and discusses various trends and challenges in the field. Specifically, [42] conducted experiments to find appropriate settings for these parameters when applying evolutionary algorithms to a multi-objective problem class. They concluded that determining the value of the scaling factor can be difficult and is highly dependent on the specific problem. Considering these findings, initial tests were conducted to define the parameters used in our study. In line with the work of [42], the initial percentage of Crossover was delimited to [0.8, 1.0], following the standard proposal for non-separable problems like the one tackled in our research. EvoImp employs a crossover for 80% of the individuals using an n-point crossover operator [43], as shown in Fig 1(f). It is also consonant with the work of [44].

The mutation process is performed on 20% of the individuals chosen randomly, except for the best one. For each individual to be mutated, the imputed value is exchanged for a candidate value. The mutation is applied only to genes that contain missing values. To accomplish this, each attribute in the dataset has a set of solutions, as shown in Table 2. This set is formed by considering all possible response options for that attribute in the evaluated dataset.

Table 2a displays a toy dataset containing five records and four attributes: "Year", "Gender", "Age", and "Have Credit". Some values in the dataset are missing and are represented by "?". Table 2b lists the possible values for each attribute. For example, the "Year" attribute can have values 1998, 2005, or 2010; and the "Gender" attribute can have values M or F. The same reasoning is applied to the other attributes.

Lobato et al. [39] adopted this technique to initiate the first MOGAImp population. The mutation operator was not implemented in MultImp. The lack of it caused a premature convergence, limiting the method's robustness. That operator is one of the main differences between MultImp and EvoImp. In other words, the proposed method implements a strategy to avoid local minimum.

The algorithm's search and optimization process occurs over predetermined generations. The population goes into a growth phase, starting with the number of MI methods adopted in

Table 2. Candidate solutions for each attribute used for the mutation process.

Year	Gender	Age	Have Credit
2010	M	25	?
?	F	?	Yes
2005	?	32	?
1998	M	?	Yes
?	?	30	No

(a) Toy Example of a dataset with MV.

Attribute	Values
Year	1998, 2005, 2010
Gender	M, F
Age	25, 30, 32
Have Credit	Yes, No

(b) Set of possible values

https://doi.org/10.1371/journal.pone.0297147.t002

the population initialization and increasing by its cross-over. This strategy aims to provide population diversity. In the second phase, the population is gradually reduced, achieving the same initial population size, allowing the analysis to choose the best solution qualitatively.

Fitness function

As mentioned earlier, the method was evaluated on an MLC scenario. For this, EvoImp performs a classification process on each individual. The goal is to analyze the performance of the classifier and, consequently, the data imputation efficiency.

Three performance measures were adopted to evaluate the classifier, as with MultImp: Exact Match, Accuracy, and Hamming Loss. The notation used by [15, 45] were adopted to describe these measures: (i) n: number of instances in the test set; (ii) q: number of labels; (iii) Y_i: set of original labels, for instance, i; and (iv) Z_i: set of predictive labels, for instance, i.

- ***Exact Match*** calculates, using a binary system, whether all the instance labels are predicted correctly. This measure, as expressed in Eq 1, is assumed to be trivial because it ignores partial predictions:

$$EM = \frac{1}{n}\sum_{i=1}^{n} I(Y_i = Z_i) \tag{1}$$

- ***Accuracy*** is also a measure that counts the correctly predicted labels of an instance. In this case, partial predictions are taken into account. Eq 2 expresses the mathematical model of this measure:

$$ACC = \frac{1}{n}\sum_{i=1}^{n} \frac{Y_i \cap Z_i}{Y_i \cup Z_i} \tag{2}$$

- ***Hamming loss*** is a measure that, in contrast to *accuracy*, evaluates the classifier's performance by finding the average of incorrect predictions. Eq 3 describes this measure:

$$HL = \frac{1}{n}\sum_{i=1}^{n} \frac{Y_i \Delta Z_i}{q} \tag{3}$$

These measures were used in lexicographical order; in other words, this approach prioritizes all the problem's objectives and then tries to satisfy them, keeping a list of priorities [46]. Thus, the fitness (f) for the problem solution can be expressed as Eq 4:

$$f = [f_0, f_1, ..., f_{n-1}] \in \mathbb{R}^n \tag{4}$$

where n is the number of objectives defined; f_n is an optimization goal. Given two fitness evaluations f_1 and f_2 and a precision threshold t, the lexicographic relation between them (noted as $\prec_l$ and $\preceq_l$) can be defined [47]:

$$\begin{aligned} f_1 \prec_l f_2 \Leftrightarrow \exists k \in [0, n_0) \cap \mathbb{N} : f_1^k < f_2^k \wedge |f_1^k - f_2^k| \geqslant \mathrm{t} \\ \wedge |f_1^i - f_2^i| < t \ \forall i < k \end{aligned} \tag{5}$$

$$f_1 =_l f_2 \Leftrightarrow |f_1^i - f_2^i| < t \ \forall i \in [0, n_o) \cap \mathbb{N} \quad (6)$$

$$f_1 \preceq_l f_2 \Leftrightarrow f_1 \prec_l f_2 \ \vee f_1 =_l f_2 \quad (7)$$

As can be observed, the Eq 5 shows $f_1 \prec_l f_2$, which means that f_1 is lexicographically less than f_2. This relationship is established when there exists an index k in in the range $[0, n_0) \cap \mathbb{N}$, such that $f_1^k < f_2^k$, indicating that the k-th component of f_1 is less than the k-th component of f_2. Additionally, the difference between f_1^k and f_2^k is greater than or equal to t. This ensures that the k-th components differ significantly by at least t. Finally, the absolute differences between corresponding components f_1^i and f_2^i should be less than t for all i less then k. In essence, this relation means that f_1 is superior to f_2 in terms of some objectives. The Eq 6 determines equality in lexicographical order ($f_1 =_l f_2$). This occurs when the absolute differences between corresponding components f_1^i and f_2^i are all less than t for all i in the range $[0, n_0) \cap \mathbb{N}$. In other words, f_1 and f_2 are considered equal regarding their performance across objectives. Finally, the Eq 7 presents $f_1 \preceq_l f_2$, which means that f_1 is either less than or equal to f_2 in lexicographical order. It combines the $\prec_l$ and $\preceq_l$ relations, indicating that f_1 is either better than or equal to f_2 in terms of the defined objectives.

These equations are used to rank and compare solutions or fitness evaluations in optimization problems, considering the objectives, prioritization, and performance. The lexicographical order approach allows for precise, multi-objective optimization when there are multiple criteria or objectives to be considered. Once the threshold t has been introduced, this formulation differs from the pure mathematical lexicographic relation. It permits the decision maker to choose the precision to compare two fitness functions. This relation allows the ranking of solutions of EvoImp as follows:

1. The EM behavior is evaluated;
2. If two or more individuals match their respective scores, the ACC evaluation is checked;
3. If the tie remains, the HL evaluation is used.

This approach allows different performance measures to be added to a single evaluation [45]. It is similar to the classical lexicographical approach, but once evolutionary algorithms are adopted, local optima can be avoided [47].

The EvoImp algorithm

As shown in Algorithm 1, EvoImp begins the execution by creating and evaluating individuals for the initial population. The datasets are initially imputed using simple imputation methods: KNNI, CMC, MC, KMI, and WKNNI (lines 1–5). Afterward, the population is evaluated and ranked based on each individual's performance (line 6). The algorithm applies the genetic operators if the stopping criterion is not attained (e.g., the number of generations).

Algorithm 1: EvoImp

```
Input: datasets with MV and parameters (see Table 4)
Output: complete datasets
1 foreach Simple Imputation Method do
2   Generate a new Individual: individual;
3   Evaluate the individual;
4   Add individual to the Current Population: currentPop ←
    individual;
5 end
6 Order currentPop using Lexicographical order;
7 while Stop criterion not reached do
```

```
8   Add to the Current population the Best Individual: currentPop ←
    bestIndividual;
9   while currentPop < Number individuals of the new generation do
10    Select Parents;
11    Appy Crossover;
12    Evaluate the new Individual;
13    Add the Individual to Current population: currentPop ←
      individual;
14  end
15  while Number of mutated individuals < 20% of Individuals of the
          new generation do
16    Randomly choose an individual from the Current population;
17    Apply Mutation;
18    Evaluate the Individual;
19    Add the Individual to Current population: currentPop ←
      individual;
20  end
21  Order currentPop using Lexicographical order;
22 end
23 return bestIndividual;
```

The elitist individual is always passed on to the next generation (line 8). The selection is performed using the tournament selection operator (line 10). Two individuals are randomly drawn in this process. These two parents exchange genetic material using a crossover operator. These steps are repeated until the population is complete. Afterward, the mutation follows the established rate (lines 15–20). The new population is arranged, and the iterative process continues until the stopping criterion is reached. The return of the algorithm is the individual that achieves the best performance (line 23).

In summary, EvoImp adopts the configuration for the parameterization of MultImp [15], except for the mutation operator, as pointed out earlier. Besides, we also corrected bugs and optimized the code, bearing in mind maintainability and reuse. Moreover, we implemented the lexicographic strategy and expanded the computational tests, expanding the technical-scientific contribution of the present work.

Computational experiments

Datasets

The experiments were designed using six multi-label datasets from the UCI Machine Learning repository (https://archive.ics.uci.edu/). The quantity datasets agree with the literature review conducted by [17], which mapped 48 papers related to experiments in the context of data imputation. Chiu's work [17] shows that most papers (77%) use up to six datasets in experiments. Another interesting finding of Chiu et al. [17] is that the UCI Machine Learning Repository is the most used. Regarding the characteristics of the datasets, most use small-scale datasets, which contain fewer than 15 attributes and 800 instances. Table 3 shows the datasets used and their features.

Regarding multi-label datasets, the works of [35, 48] must be mentioned. These studies, as well as EvoImp, used datasets obtained at the UCI repository and formatted using the Mulan library (http://mulan.sourceforge.net/). The datasets used in these papers have similar characteristics (cardinality, density, and the number of instances) to those chosen in this paper. This observation highlights the experimental setup consonance with the state of the art and the EvoImp potential applicability in real-world problems.

Table 3. Datasets used in experiments.

Dataset	Cod.	Domain	Inst.	Nominal Atr.	Numerical Atr.	Total Atr.	Labels	Cardinality	Density
Birds	b	Audio	645	2	258	260	19	1.014	0.053
Cal500	c	Music	502	0	68	68	174	26.044	0.150
Emotions	e	Music	593	0	72	72	6	1.869	0.311
Flags	f	Image	194	9	10	19	7	3.392	0.485
Scene	s	Image	2407	0	294	294	6	1.074	0.179
Yeast	y	Biology	2417	0	103	103	14	4.237	0.303

https://doi.org/10.1371/journal.pone.0297147.t003

Experimental setup

In the experiments, the missing values were artificially added to each dataset with the following rates: 5%, 10%, 15%, 20%, 25%, and 30%. This "amputation" process was carried out using the MCAR mechanism, as described in Santos (2019) [49]. The complete experimental configuration consisted of 36 datasetss with missing data, and these datasetss underwent a comparative evaluation. This evaluation involved five simple imputation methods: KNNI, CMC, MC, KMI, and WKNNI.

The following classification methods were used for the multi-label learning tasks: Binary Relevance (BR), Hierarchy of Multi-label classifiER (HOMER), Multi-Label K-Nearest Neighbors (ML-KNN), Classifier Chains (CC), and Ensembles of Classifier Chains (ECC) [21, 50]. K-fold cross-validation was used for the classification model's evaluation (learning and testing). Table 4 summarizes the overall parameters which were used in the experiments.

Regarding the simple imputation methods, the parameters recommended by [7] were used. The mutation rate (MR) chosen is higher than the typical usage rates because the starting point is not random. Therefore, considering that the initial population is obtained by other methods, parameterization experiments demonstrated that a higher MR yields better results, providing fast convergence. The entire experimental setup and the obtained results are available as supplementary material on the project's GitHub (https://github.com/jacobjr/EvoImp).

Implementation

The GA was programmed in the Java language, version 8.1, based on the works of [15, 44]. Other components used third-party implementations as follows:

Table 4. Parameter settings used in the experiments.

Parameter	Value
Initial population	five individuals (imputed datasets)
Generations	7
Crossover rate	80% of individuals
Mutation rate	20%
Selection type	Tournament (size = 2)
Imputation methods	KNNI, CMC, MC, KMI, and WKNNI
MV rates	5, 10, 15, 20, 25, and 30%
Method of MV occurrence	MCAR
MLC algorithms	BR, HOMER, ML-KNN, CC, and ECC
K-fold cross-validation	k = 10

https://doi.org/10.1371/journal.pone.0297147.t004

- For the multi-label classifiers, Mulan's library (https://mulan.sourceforge.net/) was used [51]. This library also contains some classifiers implemented in Weka (https://www.cs.waikato.ac.nz/ml/weka/index.html) [52].
- The simple imputation methods used for forming the first population of EvoImp and in the comparative analyses are implemented in KEEL-software (http://www.keel.es/) [53].

It is noteworthy that GA used in the EvoImp was fully implemented by the authors despite KEEL providing a framework for evolutionary computation. This design decision aimed to give us more control over the experiments. The computational complexity is another crucial aspect to consider in implementing this proposed method. It plays a vital role in determining the feasibility and efficiency of applying bio-inspired techniques to solve optimization problems. Addressing this concern and reducing computational complexity enhances the algorithm's applicability and scalability. As a result, it makes it more suitable for handling larger datasets and complex optimization landscapes, particularly in multi-label classification tasks. More detailed information about EvoImp's computational complexity can be found in the supplementary materials on the project's GitHub repository.

Results and analysis

This section examines the results obtained from the computational experiments. The data displayed in the following tables show the differences in performance between the methods for each percentage of missing values analyzed (5%, 10%, 15%, 20%, 25%, and 30%). The best results are highlighted in bold for easy viewing. The metrics (Exact Match (↑), Accuracy (↑), and Hamming Loss (↓)) are presented with these symbols, where (↑) indicates that higher values reflect better performance, and (↓) indicates that lower values represent better performance.

Binary relevance

In the learning performed with the BR classifier, the results showed that the EvoImp was numerically superior (Table 5). In the EM evaluation, EvoImp outperformed its competitors in 35 of the 36 datasets evaluated (97.22%). The proposed method demonstrated superior performance compared to others in 18 scenario datasetss (50%) regarding the Accuracy evaluation measure. Finally, considering the HL, EvoImp outperformed the baseline methods in 16 datasets (44.44%).

It is essential to highlight the priorities adopted in the EvoImp lexicographic order, prioritizing the evaluation with EM, as mentioned in the Subsection "Fitness Function", which explains the performance decrease for the ACC and HL metrics considering the binary relevance classifier.

Hierarchy Of Multi-label Classifier (HOMER)

The results for the HOMER classifier are given in presented in Table 6. Analyzing the results, it is possible to observe that EvoImp is also superior to the others in 35 of the 36 datasets used in the experiments (97.22%) regarding the EM metric. These results corroborate the ones obtained from the Binary Relevance classifier.

Continuing analyzing Table 6 results, regarding the ACC evaluation measure, EvoImp outperformed the baseline methods in 23 datasets (63.88%). The HL results show that EvoImp had the slightest error in classification in 19 out of 36 datasets (52.78%). In summary, EvoImp outperformed the methods for all performance measures for HOMER classifier, in consonance with the results for BR classifier as well.

Table 5. Experimental results for the binary relevance classifier.

		Exact Match (↑)						Accuracy (↑)						Hamming Loss (↓)					
%[1]	Db[2]	KMI[3]	KNNI[3]	MC[3]	CMC[3]	WKNNI[3]	EvoImp	KMI	KNNI	MC	CMC	WKNNI	EvoImp	KMI	KNNI	MC	CMC	WKNNI	EvoImp
5	b	46.67	48.54	51.32	51.22	51.31	**52.57**	58.01	61.82	62.33	60.91	62.32	**62.94**	05.52	05.21	05.05	05.06	05.13	**05.01**
	c	33.77	34.92	33.93	34.13	34.86	**35.24**	44.04	44.88	44.87	**45.12**	44.75	44.73	15.81	15.92	15.55	**15.42**	15.91	15.87
	e	50.78	50.82	48.83	51.21	50.71	**51.44**	54.04	56.74	53.42	56.77	56.54	**57.24**	24.92	25.02	25.94	24.78	25.32	**24.51**
	f	58.82	58.88	60.22	59.41	58.44	**61.94**	**70.02**	68.75	69.82	69.51	68.63	68.48	**27.42**	28.49	27.86	28.03	28.63	27.76
	s	53.59	56.92	51.04	52.81	56.93	**58.12**	51.23	55.85	48.84	50.71	55.27	**56.68**	14.14	13.34	14.67	13.93	13.31	**13.02**
	y	50.67	49.72	50.17	50.53	50.64	**51.32**	59.90	59.92	60.33	61.17	59.94	**61.78**	24.83	24.96	24.22	23.92	24.89	**23.84**
10	b	46.81	48.43	47.71	46.82	46.33	**49.87**	56.51	60.02	**60.24**	56.93	57.42	60.11	05.74	05.32	**05.12**	05.44	05.51	05.32
	c	35.54	36.14	34.71	34.63	36.46	**36.63**	44.45	45.62	45.36	45.22	45.78	**45.85**	15.52	15.53	**14.92**	15.04	15.52	15.51
	e	47.54	49.87	44.82	49.63	48.25	**50.97**	53.62	53.24	47.04	52.27	**54.43**	54.12	26.72	26.05	28.13	**25.22**	26.56	25.96
	f	57.52	60.22	57.43	58.41	60.34	**61.64**	70.43	73.12	70.99	68.73	73.34	**73.72**	26.93	26.13	26.22	28.31	25.96	**25.22**
	s	52.65	56.81	47.68	51.82	57.03	**58.34**	51.12	56.13	43.61	48.16	56.56	**57.37**	13.94	12.94	15.25	13.91	12.85	**12.34**
	y	49.56	50.64	48.81	48.42	50.22	**51.93**	59.23	60.31	59.66	59.13	60.37	**61.17**	24.85	24.59	**24.21**	24.42	24.67	24.43
15	b	44.42	45.87	47.32	43.86	44.62	**47.36**	56.31	57.38	59.03	54.58	54.52	**59.14**	05.63	05.26	04.98	05.62	05.31	**04.92**
	c	34.73	35.41	36.06	36.01	35.95	**36.17**	43.91	43.97	47.03	**47.55**	44.72	47.36	14.92	15.43	14.19	**14.04**	15.32	14.08
	e	48.41	49.43	44.09	47.61	48.13	**50.32**	52.73	55.32	46.63	53.51	55.74	**56.22**	26.31	25.59	27.05	25.13	**24.92**	25.46
	f	61.74	61.76	59.03	61.85	63.71	**64.35**	73.11	73.45	72.14	72.16	74.95	**75.07**	25.91	24.46	25.95	24.98	23.06	**22.87**
	s	50.72	58.16	46.35	49.64	58.19	**58.95**	48.94	**57.95**	41.51	47.15	57.14	57.65	14.24	12.56	15.41	13.54	12.51	**12.21**
	y	47.50	50.94	48.71	49.95	50.94	**51.44**	57.61	60.25	60.27	60.76	59.91	**61.09**	24.41	24.34	23.36	**23.31**	24.74	24.38
20	b	43.04	45.84	43.28	42.51	43.79	**47.37**	54.66	**58.27**	53.94	52.26	55.51	57.65	05.51	**04.86**	05.15	05.52	05.20	04.97
	c	35.82	35.41	35.55	35.47	35.45	**36.45**	44.02	43.84	**47.36**	46.38	43.43	44.31	14.44	15.17	**13.51**	13.64	15.27	14.94
	e	**50.12**	45.44	40.82	48.46	46.23	48.92	**52.38**	50.34	39.94	51.17	50.49	51.24	25.36	27.28	27.51	24.04	27.42	**23.85**
	f	57.93	61.05	59.61	62.51	60.25	**63.37**	70.02	71.76	71.74	71.73	**72.87**	72.12	27.14	24.46	26.04	24.97	**23.92**	24.56
	s	45.52	57.74	43.61	47.48	58.45	**58.74**	42.01	57.26	39.08	44.34	57.63	**57.73**	15.23	12.36	15.48	13.69	**11.85**	11.93
	y	49.25	50.92	50.16	49.23	51.57	**51.78**	58.53	61.06	61.67	**61.75**	60.85	60.82	24.52	24.07	**22.01**	22.36	23.92	23.97
25	b	43.45	43.44	44.15	42.14	43.94	**44.75**	55.47	56.25	56.56	52.57	**57.78**	56.91	05.15	05.01	04.84	05.01	04.83	**04.73**
	c	37.76	37.21	37.75	36.26	36.47	**37.85**	45.81	44.85	50.56	47.77	44.21	**50.56**	13.94	14.98	12.64	13.01	15.06	**12.62**
	e	43.23	45.14	38.22	46.57	45.06	**47.72**	44.35	**50.13**	37.21	49.32	49.41	50.12	25.94	26.82	26.31	**23.48**	27.09	25.24
	f	60.25	58.92	62.23	60.24	59.71	**63.46**	72.51	72.65	73.13	70.32	72.47	**73.04**	27.04	25.83	26.08	27.42	**25.15**	25.31
	s	47.14	59.36	39.94	45.16	58.87	**60.16**	43.34	58.68	36.53	42.32	58.74	**59.14**	14.61	11.73	16.02	13.61	12.09	**11.72**
	y	49.72	51.75	48.51	49.21	51.21	**51.96**	**62.96**	61.65	61.74	61.71	61.08	61.74	**21.42**	23.21	22.05	21.84	23.68	23.29
30	b	39.15	39.41	42.27	42.31	41.03	**43.36**	50.25	51.71	52.82	52.76	**53.48**	53.03	05.51	05.48	**04.94**	05.22	05.21	05.14
	c	37.13	35.92	37.82	37.65	35.43	**38.01**	45.21	43.24	**49.92**	49.01	43.26	47.57	13.62	14.73	**12.02**	12.21	14.62	12.97
	e	44.52	45.96	41.12	48.91	48.03	**49.62**	48.63	49.22	38.85	53.67	53.65	**54.37**	26.52	26.81	26.24	24.06	26.13	**23.81**
	f	62.04	62.42	59.71	64.74	63.11	**65.54**	74.32	74.23	74.74	75.13	74.58	**75.46**	24.68	24.42	24.84	23.72	24.73	**23.26**
	s	44.53	59.32	39.46	44.18	58.97	**59.62**	40.94	59.01	35.78	41.92	**59.38**	59.23	15.37	11.79	15.32	13.52	11.74	**11.62**
	y	50.13	51.82	49.71	49.04	51.57	**52.71**	59.96	62.23	62.72	**63.02**	62.04	62.84	23.52	22.83	**20.92**	21.04	22.94	22.52
Avg rank	-	5	2	6	4	3	**1**	6	3	5	4	2	**1**	6	5	3	2	4	**1**

[1]"%" refers to the percentage of missing data analyzed (5%, 10%, 15%, 20%, 25%, and 30%).

[2]"Db" refers to the datasets used in the experimental setup, and these letters' abbreviations can be found in Table 3.

[3]Acronyms are related to each data imputation method tested, listed in S1 Table. Abbreviations.

https://doi.org/10.1371/journal.pone.0297147.t005

Table 6. Experimental results for HOMER classifier.

		Exact Match (↑)						Accuracy (↑)						Hamming Loss (↓)					
%[1]	Db[2]	KMI[3]	KNNI[3]	MC[3]	CMC[3]	WKNNI[3]	EvoImp	KMI	KNNI	MC	CMC	WKNNI	EvoImp	KMI	KNNI	MC	CMC	WKNNI	EvoImp
5	b	43.83	48.63	49.72	48.13	49.21	**52.47**	54.05	59.51	58.92	56.42	60.03	**60.72**	06.62	05.96	05.84	06.12	06.25	**05.58**
	c	36.43	36.41	35.36	36.48	36.54	**37.93**	35.01	35.64	34.42	34.74	35.21	**35.56**	20.42	**20.21**	20.52	20.45	20.47	20.39
	e	47.12	51.04	46.72	51.12	49.37	**53.05**	51.92	54.96	50.70	55.73	53.62	**56.31**	26.64	25.62	26.43	25.28	26.41	**24.85**
	f	61.32	61.53	60.69	60.02	60.23	**63.02**	68.25	67.72	67.71	66.04	67.82	**68.42**	27.33	27.83	27.72	29.12	27.51	**27.28**
	s	51.92	54.47	50.43	51.72	55.04	**55.32**	48.78	52.66	47.95	48.83	53.12	**53.35**	14.90	14.26	15.22	14.62	14.04	**14.01**
	y	50.42	50.12	48.97	50.04	50.79	**50.92**	58.24	58.63	57.12	56.94	**58.82**	58.64	25.98	25.70	26.53	26.41	**26.02**	26.13
10	b	44.92	46.96	45.29	43.16	47.72	**48.25**	55.32	56.41	57.09	52.73	57.26	**57.74**	06.52	06.42	06.15	06.43	06.12	**06.04**
	c	35.72	36.84	35.93	36.92	36.92	**37.95**	34.33	36.38	34.62	34.95	36.03	**36.75**	20.03	19.62	19.66	19.62	19.71	**19.52**
	e	48.9	48.83	43.74	49.54	49.62	**50.64**	53.46	53.32	47.43	52.21	53.74	**53.95**	**26.21**	26.72	27.06	26.83	26.68	26.32
	f	60.74	60.32	58.93	60.74	60.48	**61.23**	68.94	71.52	67.73	67.71	**69.92**	67.87	27.42	**25.93**	27.64	27.92	27.01	27.74
	s	52.12	55.23	47.05	47.79	55.38	**55.72**	49.44	53.23	43.18	44.08	**53.56**	53.53	14.35	13.78	16.02	15.64	13.82	**13.71**
	y	50.54	50.52	50.31	48.95	50.03	**53.98**	**58.82**	58.01	58.44	57.58	57.63	58.44	25.72	25.72	**25.03**	25.51	26.21	25.87
15	b	42.13	45.62	47.24	43.32	44.68	**47.34**	50.22	56.01	57.65	51.63	56.01	**57.88**	06.83	06.11	06.07	06.82	06.14	**06.02**
	c	36.82	37.64	36.33	35.61	37.02	**37.73**	36.22	36.94	34.85	34.47	36.72	**37.23**	**18.52**	18.91	19.12	19.13	18.91	18.72
	e	47.43	47.52	42.24	48.12	45.75	**49.34**	51.92	50.11	44.42	51.58	49.59	**52.52**	27.31	26.82	26.56	25.42	26.81	**24.71**
	f	60.72	62.41	61.01	59.22	61.73	**63.98**	70.24	72.93	69.14	68.42	**73.13**	72.84	25.72	23.87	26.82	26.73	23.71	**23.62**
	s	50.91	56.42	44.45	48.42	56.41	**57.04**	47.86	54.52	40.31	45.15	54.98	**55.83**	14.72	13.61	16.02	14.47	13.42	**13.36**
	y	47.32	50.34	49.02	49.94	50.98	**51.64**	54.23	57.91	58.27	57.31	**58.82**	58.31	25.47	25.76	**24.62**	25.34	25.52	25.67
20	b	42.62	44.24	44.38	43.27	43.93	**45.14**	52.07	52.93	54.32	51.74	**55.18**	54.72	06.34	06.12	**05.74**	06.34	05.92	05.94
	c	35.72	37.34	34.92	35.34	37.25	**37.53**	34.92	36.48	34.45	34.62	36.35	**36.52**	18.24	18.46	**18.14**	18.17	18.32	18.36
	e	46.75	49.13	40.01	46.85	47.69	**49.68**	48.54	51.41	39.54	51.15	49.05	**52.37**	27.23	26.32	27.14	**25.34**	27.32	26.04
	f	58.62	60.54	56.72	62.18	60.48	**64.54**	67.82	70.31	66.97	69.11	**70.44**	69.93	27.62	25.44	27.48	25.13	25.52	**24.27**
	s	46.67	56.72	42.96	47.32	56.03	**57.54**	42.73	55.52	38.31	43.92	55.88	**56.47**	15.44	12.93	16.01	14.12	13.15	**12.62**
	y	49.07	51.62	48.57	49.16	50.82	**51.84**	57.13	**59.67**	56.78	57.12	58.46	59.44	25.65	24.72	**24.54**	24.82	25.21	24.88
25	b	43.62	41.95	43.56	40.18	41.81	**44.98**	54.43	54.34	55.82	52.53	55.78	**58.04**	05.62	05.74	05.56	05.78	05.92	**05.42**
	c	36.02	36.92	36.63	36.18	37.55	**37.79**	37.32	37.93	34.88	35.04	38.16	**38.43**	16.62	17.58	17.62	17.51	17.34	**17.35**
	e	37.34	46.42	39.28	46.32	45.17	**47.79**	35.23	46.87	37.72	47.64	47.25	**49.48**	30.53	27.82	27.01	25.87	28.23	**25.28**
	f	59.64	59.78	62.03	59.82	59.64	**63.63**	68.11	70.02	68.18	68.53	**70.35**	68.81	28.92	26.23	28.28	27.49	**25.81**	27.73
	s	47.34	55.52	41.28	46.72	55.56	**55.73**	44.01	**55.02**	38.22	44.19	54.92	55.06	14.72	**13.26**	15.32	13.64	13.38	13.39
	y	49.12	51.54	48.92	49.13	**51.52**	51.15	57.92	60.03	57.18	57.42	59.92	**60.21**	**23.88**	24.33	24.22	24.01	24.38	24.13
30	b	38.35	41.72	40.84	40.48	40.52	**42.93**	50.44	52.92	51.75	48.73	51.21	**51.68**	06.22	06.28	05.73	06.24	06.58	**05.43**
	c	35.67	36.68	36.13	35.54	36.26	**36.88**	37.55	37.74	34.98	34.32	37.05	**37.78**	**16.16**	17.04	16.46	16.87	17.22	17.06
	e	43.54	46.78	40.64	49.26	48.12	**50.04**	46.26	50.02	38.34	**53.35**	52.88	51.42	27.05	26.92	26.36	**24.34**	25.88	25.42
	f	61.14	65.26	63.54	63.02	64.12	**66.24**	71.45	74.88	72.73	74.34	74.15	**75.48**	25.72	22.76	24.94	23.96	24.02	**22.24**
	s	44.46	54.92	38.34	43.68	55.44	**56.45**	42.03	55.12	33.86	41.32	55.04	**56.25**	15.48	12.92	16.26	14.12	13.07	**12.75**
	y	49.85	52.04	48.68	50.76	52.62	**52.87**	59.92	**61.84**	59.86	60.11	61.52	61.55	23.86	23.42	22.44	**22.02**	23.36	23.24
Avg rank	-	5	2	6	4	3	**1**	4	3	6	5	2	**1**	6	2	5	3	4	**1**

[1]"%" refers to the percentage of missing data analyzed (5%, 10%, 15%, 20%, 25%, and 30%).

[2]"Db" refers to the datasets used in the experimental setup, and these letters' abbreviations can be found in Table 3.

[3]Acronyms are related to each data imputation method tested, listed in S1 Table. Abbreviations.

https://doi.org/10.1371/journal.pone.0297147.t006

Multi-Label k-Nearest Neighbors

The results obtained with the ML-KNN classifier is shown in Table 7. As can be seen, EvoImp showed similar performance to the previous scenarios considering the BR and the HOMER classifiers. For instance, considering the primary analyzed metric (EM), EvoImp outperformed the baseline methods at 97.22%. Considering the ACC and HL, the EvoImp presented superior performance for 20 (55.55%) and 22 (61.11%) datasets, respectively.

Classifier Chains

The results for the Classifier Chains are presented in Table 8. Again, EvoImp outperformed the baseline methods for all evaluation measures considered: EM with superiority in 32 out of 36 datasets (88.88%), ACC with 30 (83.33%), and HL with 22 (61.11%).

Ensembles of Classifier Chains

The last scenario analyzed was considering the Ensemble Classifier Chains method. The results are shown in Table 9. The results obtained with the ECC (Table 9) also show a significant advantage of EvoImp over competitors in the analyses performed. However, EvoImp had the lowest performance, with numerical superiority in 29 (80.55%) datasets in the evaluation with EM, 16 (44.44%) for ACC, and 17 (47.22%) for HL.

In summary, the EvoImp performance for the ECC presents the same pattern described in the previous scenarios, demonstrating the EvoImp robustness.

Discussion

In summary, EvoImp proved to be competitive in all classification scenarios, which underlines the fact that the optimization of imputation through evolutionary strategies, such as genetic algorithms, is an excellent alternative for handling missing values in the preprocessing phase of data analysis. It should be noted that the algorithm created performed optimizations based on simple imputation methods (applied to the initial population of EvoImp). Considering the computational experiments, other factors should be highlighted regarding the EvoImp performance:

- **Maximizing the labels' success**: The primary purpose of classification, particularly in this study, is the correct labeling of data instances, a task that is becoming increasingly complex in the multi-labeling scenario. In the EM measure, where the classifier must label all the classes of an instance correctly so that they can be counted correctly, the proposed method achieved better performance in 92.22% of all the datasets in all the scenarios. This performance is more evident in BR, HOMER, and ML-KNN, with 35 out of the 36 datasets. Another measure that allows this conclusion is ACC. The superior performance achieved by the EvoImp is more apparent in the analyses with the CC and HOMER classifiers (with 30 and 23 datasets, respectively). In general terms, EvoImp was better in 68.3% of all used datasets. This can be explained by the fact that this measure is flexible regarding the number of successes achieved by labels. For example, if an instance belongs to five labels and obtains four correct labels, it achieves an 80% degree of accuracy. At the same time, the excellent performance of ACC indicates that the classifier can increase its labeling capacity. This can be confirmed by analyzing the classification error evaluated using HL. In this metric, the proposed method obtained the lowest error (53.33%). It is worth mentioning that the results obtained reflect the lexicographic order chosen (as explained in subsection "Fitness function"), demonstrating the method's superiority over all the others. A comparison shows that when ACC increases, there is an automatic reduction in the HL error, justifying the usage of

Table 7. Experimental results for the ML-KNN Classifier.

		Exact Match (↑)						Accuracy (↑)						Hamming Loss (↓)					
%[1]	Db[2]	KMI[3]	KNNI[3]	MC[3]	CMC[3]	WKNNI[3]	EvoImp	KMI	KNNI	MC	CMC	WKNNI	EvoImp	KMI	KNNI	MC	CMC	WKNNI	EvoImp
5	b	46.52	50.44	47.14	49.82	49.04	**51.43**	60.32	59.74	58.76	60.52	59.67	**60.76**	04.62	04.74	04.76	**04.52**	04.74	04.78
	c	36.06	35.62	36.24	36.46	36.08	**36.64**	61.22	61.40	61.59	62.26	60.61	**62.28**	13.42	13.54	**13.11**	13.23	13.52	13.24
	e	57.64	57.23	54.64	56.93	57.74	**59.32**	67.71	69.43	64.18	67.11	**69.52**	69.43	19.11	18.72	19.73	19.25	18.97	**18.35**
	f	61.42	61.61	61.25	60.98	61.82	**62.44**	70.63	70.91	**73.26**	72.42	71.21	71.54	27.72	27.31	**26.44**	26.72	27.38	27.13
	s	63.74	67.82	58.57	64.02	68.08	**68.13**	65.67	71.72	57.88	66.26	**72.15**	**72.15**	09.13	07.94	10.32	08.84	**07.92**	**07.92**
	y	56.24	57.88	55.96	55.62	58.03	**58.21**	72.46	72.92	**73.16**	72.85	72.66	72.62	19.21	**18.78**	18.82	18.94	**18.78**	**18.78**
10	b	47.02	46.04	44.28	43.93	46.07	**47.92**	55.94	**57.62**	55.53	56.24	56.88	57.04	04.62	04.64	04.66	**04.51**	04.68	04.62
	c	36.14	36.78	37.66	36.92	36.64	**37.72**	61.51	61.88	65.53	65.44	61.89	**65.66**	13.01	13.22	**12.46**	12.52	13.21	**12.46**
	e	55.46	55.18	50.82	56.06	54.07	**56.96**	65.62	64.05	56.69	65.92	64.06	**66.67**	20.26	20.62	21.87	19.72	20.96	**19.62**
	f	59.94	60.26	59.42	62.37	59.48	**63.11**	72.85	72.48	70.82	73.64	72.05	**73.86**	26.42	26.78	27.61	25.12	27.04	**24.92**
	s	60.92	69.46	48.92	59.88	69.72	**69.87**	62.11	74.82	43.89	61.34	**74.96**	**74.96**	09.52	07.38	11.92	09.56	**07.32**	**07.32**
	y	56.23	58.14	54.29	54.92	58.14	**58.46**	73.02	73.24	**73.38**	73.02	73.16	73.22	18.78	18.22	18.78	18.46	18.24	**18.12**
15	b	45.62	45.34	45.46	46.01	43.54	**48.16**	56.24	57.58	55.24	57.26	56.34	**60.36**	04.51	04.42	04.54	**04.36**	04.52	**04.36**
	c	36.52	37.74	**39.56**	39.02	37.74	**39.56**	62.22	61.01	67.28	67.09	61.46	**67.47**	12.62	12.74	**11.64**	11.72	12.76	**11.64**
	e	55.02	54.64	47.48	53.29	54.24	**55.31**	64.44	65.45	54.78	64.49	65.01	**65.86**	20.62	19.88	21.12	19.91	20.26	**19.52**
	f	60.56	61.14	56.62	58.48	60.62	**61.87**	**72.96**	72.02	72.14	70.32	71.75	72.42	**26.16**	26.62	27.34	28.18	27.12	26.41
	s	59.56	70.24	42.32	57.51	**70.56**	**70.56**	59.82	75.14	35.72	56.62	**75.38**	75.28	09.92	**07.11**	12.72	09.87	**07.11**	**07.11**
	y	53.74	59.82	55.81	55.66	59.62	**59.92**	69.65	73.19	73.81	**74.36**	73.02	73.27	19.29	**17.44**	17.92	17.87	17.56	**17.44**
20	b	43.82	45.62	45.04	46.49	44.52	**50.91**	57.56	57.14	54.97	55.62	56.65	**57.64**	**04.14**	**04.14**	04.22	**04.14**	**04.14**	04.71
	c	38.29	37.82	**40.14**	40.02	37.96	**40.14**	63.76	62.92	69.94	**70.08**	62.94	69.96	11.81	12.44	**11.05**	**11.05**	12.41	**11.05**
	e	53.42	55.14	41.06	52.07	55.26	**55.96**	59.41	63.97	42.25	61.26	64.84	**65.52**	21.21	20.18	22.79	**19.32**	20.31	20.26
	f	61.71	61.67	59.49	60.62	61.82	**63.17**	74.36	75.34	74.82	74.76	74.87	**76.26**	24.15	24.62	25.84	25.76	24.74	**23.92**
	s	54.15	71.92	39.88	54.59	71.41	**72.06**	51.82	**77.34**	34.32	52.56	77.22	77.26	10.31	**06.72**	12.56	10.01	**06.72**	**06.72**
	y	54.66	60.24	53.32	53.84	60.62	**60.76**	68.82	74.74	**76.06**	75.92	74.94	75.06	19.88	16.82	17.76	17.51	**16.72**	17.44
25	b	44.16	44.24	41.36	42.21	47.06	**47.37**	56.72	58.22	56.86	56.81	58.56	**58.82**	**03.82**	03.86	03.84	03.85	**03.82**	**03.82**
	c	39.28	39.16	**41.94**	41.25	38.79	**41.94**	63.32	62.92	70.64	69.62	62.66	**70.71**	11.52	12.16	**10.34**	10.55	12.18	**10.34**
	e	43.54	53.02	33.78	47.92	50.52	**53.64**	42.92	58.48	30.36	54.64	55.57	**59.16**	21.92	21.54	22.72	**19.74**	22.02	21.24
	f	60.56	60.32	62.24	58.76	59.84	**62.55**	75.42	75.29	74.21	**75.82**	74.95	74.36	26.39	**25.02**	25.96	26.52	25.84	25.64
	s	52.65	72.24	36.32	50.96	72.34	**72.58**	49.64	78.31	32.02	48.36	78.17	**78.46**	10.52	**06.41**	12.44	10.42	**06.41**	**06.41**
	y	54.42	61.25	54.49	54.31	61.06	**61.88**	76.76	74.91	77.14	**77.36**	75.05	75.21	17.02	**16.16**	16.82	16.91	**16.16**	16.78
30	b	42.32	42.16	38.58	42.57	39.82	**46.99**	55.11	54.46	55.34	55.26	54.25	**55.76**	**03.92**	04.01	**03.92**	**03.92**	04.08	04.06
	c	39.74	38.26	42.52	42.64	38.12	**42.71**	65.76	63.84	74.08	**73.65**	64.24	**73.65**	10.92	11.86	**09.68**	09.72	11.74	09.78
	e	50.02	52.99	31.62	52.64	54.94	**55.08**	57.56	63.01	28.56	64.88	64.86	**65.57**	20.91	20.35	22.46	19.62	19.68	**19.51**
	f	**65.72**	63.36	61.38	64.32	60.71	65.19	**79.12**	77.45	81.27	77.99	77.32	76.06	**22.61**	24.22	23.45	22.98	25.16	24.12
	s	51.32	72.26	34.62	48.84	72.26	**72.32**	49.44	78.92	31.56	45.68	**79.12**	79.04	10.62	06.26	12.18	10.44	**06.12**	**06.12**
	y	52.56	62.58	53.72	54.31	62.88	**63.09**	73.61	75.85	**79.42**	79.24	76.16	76.12	18.88	15.26	16.12	16.14	**15.14**	**15.14**
Avg rank	-	5	2	6	4	3	**1**	6	3	5	2	4	**1**	5	2	6	3	4	**1**

[1]"%" refers to the percentage of missing data analyzed (5%, 10%, 15%, 20%, 25%, and 30%).

[2]"Db" refers to the datasets used in the experimental setup, and these letters' abbreviations can be found in Table 3.

[3]Acronyms are related to each data imputation method tested, listed in S1 Table. Abbreviations.

https://doi.org/10.1371/journal.pone.0297147.t007

Table 8. Experimental results for the Classifier Chains.

		Exact Match (↑)						Accuracy (↑)						Hamming Loss (↓)					
%[1]	Db[2]	KMI[3]	KNNI[3]	MC[3]	CMC[3]	WKNNI[3]	EvoImp	KMI	KNNI	MC	CMC	WKNNI	EvoImp	KMI	KNNI	MC	CMC	WKNNI	EvoImp
5	b	49.62	48.56	49.91	50.45	50.21	**51.46**	61.52	60.81	61.44	60.99	61.62	**62.26**	05.21	05.25	05.06	05.11	05.36	**05.02**
	c	34.64	34.92	34.34	34.56	34.81	**35.34**	40.06	40.14	40.21	**40.46**	40.05	40.32	17.34	17.46	17.14	**17.01**	17.42	17.13
	e	49.82	49.91	45.86	48.82	48.81	**52.06**	55.45	55.68	52.82	53.59	54.76	**56.56**	25.72	**25.42**	27.34	26.72	25.86	25.61
	f	59.52	58.98	60.56	59.76	59.32	**62.26**	66.72	67.11	68.62	68.18	67.92	**70.06**	29.71	29.82	28.26	28.71	29.08	**26.86**
	s	56.52	59.78	51.51	54.26	58.17	**59.96**	56.82	61.84	51.42	55.46	60.36	**62.02**	14.14	13.62	15.52	13.88	14.04	**13.52**
	y	49.29	49.82	48.14	48.12	49.98	**50.36**	56.22	56.38	54.54	55.02	56.44	**56.48**	26.56	26.48	26.61	26.46	**26.42**	26.51
10	b	46.22	48.16	48.44	45.12	45.92	**49.56**	57.61	60.52	60.38	55.76	58.13	**61.62**	05.41	05.36	05.02	05.38	05.56	**04.92**
	c	**36.18**	35.56	34.82	35.08	35.31	36.06	**45.45**	42.02	40.76	40.61	41.68	41.82	**16.32**	16.52	16.31	16.36	16.72	16.56
	e	45.92	49.48	41.71	49.66	48.38	**50.92**	52.36	54.25	46.07	52.02	**55.06**	54.92	28.08	26.89	29.52	27.24	26.56	**26.44**
	f	57.16	59.62	57.74	58.33	61.05	**61.58**	69.72	71.76	71.41	68.12	72.26	**72.82**	28.18	27.12	26.76	28.81	26.08	**25.36**
	s	53.54	59.92	48.46	52.72	58.71	**60.08**	54.16	61.85	46.32	51.16	60.32	**61.96**	14.12	13.18	15.77	14.09	13.66	**13.12**
	y	48.52	49.26	47.24	46.96	49.32	**50.56**	55.34	55.69	53.82	53.89	56.92	**57.75**	26.24	26.62	26.54	26.26	26.28	**25.66**
15	b	44.52	45.38	**46.62**	41.91	44.28	**46.62**	55.16	56.21	**57.62**	52.88	55.62	57.61	05.72	05.32	05.08	05.59	05.32	**05.06**
	c	**35.82**	35.06	35.39	34.62	34.51	35.75	41.46	41.92	41.18	40.92	41.36	**42.52**	16.06	16.28	**15.62**	15.76	16.38	15.67
	e	47.96	47.92	42.16	50.31	47.88	**50.96**	50.92	**56.28**	42.06	55.24	53.76	56.08	27.12	26.16	27.42	24.54	26.76	**24.18**
	f	61.52	62.36	58.11	60.89	62.82	**64.89**	72.02	73.46	72.09	72.92	73.74	**74.19**	26.12	24.87	26.42	25.74	24.32	**23.62**
	s	51.72	59.86	47.31	50.26	59.02	**60.18**	52.04	62.42	44.26	48.61	61.02	**62.75**	14.56	12.92	15.24	13.56	13.12	**12.82**
	y	46.11	51.36	48.34	46.72	51.06	**51.82**	53.16	57.91	55.22	53.88	57.46	**58.16**	26.02	25.05	25.14	26.02	25.38	**24.96**
20	b	42.02	44.84	43.96	43.46	44.33	**46.17**	53.25	57.48	55.44	53.13	55.42	**58.14**	05.52	05.06	05.07	05.32	05.26	**04.92**
	c	35.74	36.16	34.91	34.78	35.26	**36.62**	41.26	42.34	41.85	40.38	41.34	**43.48**	15.46	15.82	**14.84**	15.21	16.16	15.32
	e	44.98	47.93	38.52	47.94	45.92	**49.38**	48.92	51.91	36.74	52.58	50.03	**53.55**	27.82	26.68	28.39	**24.91**	27.64	25.86
	f	57.62	61.38	58.97	60.76	59.02	**61.88**	68.91	70.86	70.72	70.54	**71.26**	**71.26**	28.92	**25.51**	26.75	26.62	25.66	25.57
	s	46.52	**59.77**	42.26	48.05	59.18	59.76	44.22	61.64	38.38	45.72	**61.62**	**61.62**	15.12	**12.64**	16.22	13.79	12.72	12.69
	y	48.82	49.91	46.06	45.92	51.38	**51.41**	56.42	56.56	53.08	53.16	58.51	**58.68**	25.66	25.88	25.62	25.64	24.82	**24.71**
25	b	44.13	43.42	43.15	41.28	45.42	**45.78**	54.62	55.51	55.06	51.62	58.34	**58.42**	05.06	05.21	04.75	04.92	04.74	**04.72**
	c	**37.56**	35.88	36.12	35.51	36.36	37.23	43.72	41.91	43.26	42.72	42.81	**44.86**	14.64	15.72	**14.24**	14.28	15.42	14.56
	e	42.24	46.12	37.26	45.08	46.22	**48.76**	43.02	49.81	35.92	48.76	49.31	**52.19**	26.82	27.18	27.26	27.61	27.15	**26.22**
	f	59.32	60.66	60.92	59.28	61.24	**62.35**	**72.66**	72.32	72.44	71.26	72.58	72.22	26.87	25.92	26.84	26.92	25.65	**25.61**
	s	46.75	59.94	39.62	45.96	60.12	**61.56**	43.94	63.26	36.72	42.92	62.26	**63.98**	14.65	12.34	15.82	13.76	12.52	**11.77**
	y	46.15	51.52	46.63	46.28	51.39	**51.67**	54.12	58.72	54.55	53.59	58.92	**58.97**	24.36	24.52	**24.24**	24.52	24.56	24.48
30	b	42.06	40.62	44.97	40.66	41.34	**45.49**	54.25	52.87	55.92	51.46	52.85	**56.49**	05.02	05.28	04.76	05.11	05.12	**04.64**
	c	36.83	35.05	35.42	36.58	34.92	**38.76**	44.24	41.92	43.76	44.25	40.92	**45.81**	14.02	15.26	**13.43**	13.49	15.44	13.82
	e	42.55	44.82	38.18	48.09	45.74	**48.80**	47.06	50.22	37.79	52.95	50.91	**54.42**	26.90	27.44	26.66	24.54	27.48	**24.26**
	f	61.60	61.65	59.86	62.92	63.31	**64.14**	75.42	73.26	75.18	74.82	74.19	**75.47**	24.62	26.06	24.98	24.76	25.12	**24.14**
	s	43.40	59.85	38.73	45.44	58.83	**60.15**	41.62	62.53	34.64	43.23	62.18	**63.29**	15.20	**12.02**	15.36	13.21	12.25	12.03
	y	49.77	52.86	47.40	47.27	52.56	**53.12**	57.91	60.48	54.96	55.64	60.32	**60.65**	24.10	23.23	23.35	**23.01**	23.46	23.12
Avg rank	-	4	2	6	5	3	**1**	4	2	5	6	3	**1**	6	3	4	2	5	**1**

[1]"%" refers to the percentage of missing data analyzed (5%, 10%, 15%, 20%, 25%, and 30%).

[2]"Db" refers to the datasets used in the experimental setup, and these letters' abbreviations can be found in Table 3.

[3]Acronyms are related to each data imputation method tested, listed in S1 Table. Abbreviations.

https://doi.org/10.1371/journal.pone.0297147.t008

Table 9. Experimental Results for the Ensemble of Classifier Chains.

		Exact Match (↑)						Accuracy (↑)						Hamming Loss (↓)					
%[1]	Db[2]	KMI[3]	KNNI[3]	MC[3]	CMC[3]	WKNNI[3]	EvoImp	KMI	KNNI	MC	CMC	WKNNI	EvoImp	KMI	KNNI	MC	CMC	WKNNI	EvoImp
5	b	53.72	54.43	52.75	52.79	53.61	**54.69**	64.42	65.78	64.31	63.40	**65.76**	65.32	04.23	04.26	04.27	**04.21**	04.26	04.28
	c	54.52	54.88	54.29	54.12	**55.16**	36.64	54.36	55.02	**56.89**	55.72	54.64	56.09	14.12	14.16	13.79	13.82	14.16	**13.72**
	e	55.56	56.82	52.16	58.26	56.92	**58.68**	61.92	**65.86**	59.12	64.57	63.42	63.93	20.22	19.05	20.93	**18.94**	19.76	19.18
	f	61.32	62.09	61.60	62.62	62.25	**65.68**	71.02	70.26	70.72	71.58	69.72	**72.56**	25.71	25.46	26.22	25.67	25.92	**24.84**
	s	59.92	63.93	56.14	57.92	**64.46**	**64.46**	59.82	65.84	54.83	57.32	65.84	**65.89**	09.82	**08.83**	10.30	09.82	08.87	08.88
	y	54.52	54.86	54.24	54.12	55.18	**55.39**	69.02	68.80	**69.44**	69.41	69.13	69.39	20.52	20.46	19.91	**19.82**	20.36	20.23
10	b	45.25	46.62	49.13	47.99	48.10	**49.28**	60.02	**61.97**	61.32	58.92	61.65	61.26	04.42	04.41	**04.23**	04.35	04.36	04.27
	c	54.16	**55.82**	51.96	52.98	55.72	37.51	56.16	55.54	57.22	57.58	55.46	**58.14**	13.55	13.83	**13.14**	13.18	13.86	13.33
	e	53.82	53.87	46.85	54.20	54.51	**55.35**	60.83	60.71	52.02	**61.93**	60.62	61.14	21.23	**20.92**	22.46	20.32	21.27	**20.92**
	f	61.33	61.37	61.52	62.78	60.93	**65.03**	71.51	72.28	72.92	73.33	73.09	**74.05**	26.24	25.52	**24.46**	24.84	25.82	24.83
	s	59.05	64.82	52.73	56.77	**65.26**	**65.26**	58.14	66.28	48.72	55.16	**66.87**	**66.87**	09.82	08.74	10.93	09.78	**08.42**	**08.42**
	y	54.19	55.82	51.97	52.93	54.08	**56.06**	70.07	70.32	69.49	69.72	70.17	**70.66**	19.92	19.78	19.91	19.82	19.79	**19.63**
15	b	47.24	47.39	47.82	44.18	47.33	**50.04**	58.07	**61.09**	60.81	57.66	60.48	59.64	04.41	**04.23**	**04.23**	04.37	04.28	04.25
	c	51.92	**56.09**	52.90	53.77	56.06	38.25	57.78	57.62	**60.23**	58.54	57.93	60.12	12.96	13.25	**12.37**	12.53	13.12	12.39
	e	53.02	52.78	45.13	54.54	52.02	**55.34**	59.75	**63.78**	48.52	62.46	61.01	62.49	21.35	20.03	22.67	19.68	20.72	**19.29**
	f	62.52	63.89	59.67	60.45	64.32	**67.09**	73.74	73.05	71.20	71.36	73.72	**74.48**	24.34	23.95	26.57	26.02	23.69	**22.92**
	s	56.64	64.82	49.73	53.68	64.62	**64.88**	56.32	65.56	46.72	51.20	**66.06**	65.52	10.17	**08.52**	10.96	10.04	08.69	**08.52**
	y	51.94	56.01	52.92	53.78	56.09	**56.32**	66.46	70.52	71.58	71.05	70.06	**72.08**	20.02	19.39	**19.13**	18.87	19.39	19.16
20	b	47.15	46.53	46.58	46.92	47.59	**51.07**	56.92	59.26	58.68	58.22	58.47	**59.46**	04.15	04.16	04.02	04.08	04.19	**03.96**
	c	53.82	56.36	52.78	52.46	**57.17**	39.12	58.82	57.67	61.92	62.09	57.35	**62.51**	12.26	12.88	11.66	11.64	12.98	**11.62**
	e	51.32	53.28	45.56	51.98	52.40	**54.67**	57.16	60.01	43.78	59.32	60.92	**61.16**	20.74	20.42	22.29	**19.44**	20.82	20.07
	f	59.92	62.56	61.07	61.74	62.02	**63.89**	71.45	72.49	71.63	70.67	72.42	**73.46**	26.57	24.09	25.55	26.79	24.34	**23.34**
	s	49.85	65.32	46.39	53.87	65.26	**65.37**	47.12	66.96	41.75	51.90	66.88	**66.95**	11.12	**08.23**	11.54	09.56	08.35	**08.23**
	y	53.88	56.32	52.70	52.45	57.19	**57.42**	69.36	70.97	**72.62**	72.06	70.97	71.24	19.82	19.09	**18.52**	18.67	18.92	18.80
25	b	44.15	43.63	45.42	45.77	44.26	**48.56**	59.72	**59.82**	58.79	58.36	59.54	59.55	03.72	03.77	03.76	03.75	03.82	**03.65**
	c	52.92	**57.29**	53.43	53.14	57.16	40.32	61.26	59.18	**64.93**	63.64	59.72	63.56	11.68	12.46	10.92	**10.14**	12.46	11.24
	e	44.38	51.93	38.00	50.97	50.42	**53.83**	46.62	56.54	38.08	58.62	55.46	**61.24**	22.02	22.06	22.28	18.42	22.37	**18.24**
	f	61.54	61.90	61.98	60.86	60.82	**63.44**	73.02	**74.30**	73.56	73.24	72.38	73.11	25.88	**24.02**	25.56	25.14	25.21	25.48
	s	51.36	66.55	44.27	51.42	65.76	**66.67**	48.65	68.52	40.28	50.07	68.02	**68.79**	10.52	**07.94**	11.56	09.62	08.17	07.99
	y	52.92	57.28	53.44	53.15	57.16	**57.72**	73.07	71.45	**73.42**	72.84	71.96	72.02	17.88	18.74	**17.73**	18.05	18.42	18.27
30	b	42.35	42.28	46.43	47.35	42.02	**49.86**	55.14	54.92	54.94	**58.26**	55.07	56.51	03.95	04.17	03.92	**03.86**	04.12	03.88
	c	54.82	**58.26**	53.42	53.47	57.89	40.80	63.32	58.67	**66.69**	66.12	59.77	66.26	11.02	12.28	10.24	**10.32**	12.11	10.39
	e	49.34	53.72	35.23	54.67	54.15	**55.16**	56.01	62.98	35.36	**65.25**	63.38	64.67	21.42	19.72	22.10	18.55	19.76	**18.22**
	f	**65.73**	63.65	62.47	64.82	62.89	65.64	**78.23**	75.35	76.48	76.72	74.86	77.47	22.76	23.27	24.26	22.82	24.07	**22.52**
	s	48.96	65.70	42.77	50.54	65.02	**65.76**	47.37	68.92	38.75	48.69	67.52	**68.97**	10.92	**07.65**	11.32	09.67	08.02	**07.65**
	y	54.82	58.28	53.49	53.42	57.87	**58.34**	72.92	73.19	74.92	**75.28**	73.42	73.46	18.28	17.71	16.92	**16.87**	17.76	17.62
Avg rank	-	5	2	6	4	3	**1**	6	2	5	3	4	**1**	6	3	4	2	5	**1**

[1]"%" refers to the percentage of missing data analyzed (5%, 10%, 15%, 20%, 25%, and 30%).

[2]"Db" refers to the datasets used in the experimental setup, and these letters' abbreviations can be found in Table 3.

[3]Acronyms are related to each data imputation method tested, listed in S1 Table. Abbreviations.

https://doi.org/10.1371/journal.pone.0297147.t009

lexicographical order instead of more complex approaches, such as Pareto Frontier Analysis, used to deal with conflicting measures.

- **Superior performance in datasets over different domains and sizes**: The six datasets used in the experiments can be divided in terms of i) different domains—the multi-label datasets used were related to the areas of audio (1), music (2), image (2), and biology (1); ii) their sizes—considering the number of instances and attributes, as was done by [54]. These datasets were curated to provide a robust experimental setup, simulating diverse real-world problems. It was noted that EvoImp performed superior in all the tests, proving that the method is robust on datasets of different domains and sizes.
- **Stable performance in the distribution rates of the missing values under study**: A critical evaluation of this study is related to the relationship between the missing values percentage and the performance measures. The results show that the EvoImp maintains its consistency, even with variations, which, in this study, was between 5% to 30% (with a rate of k = 5%). These rates agree with those used in most studies in the literature—one related work that addresses this discussion is [17]. A total of 48 related articles from 2011 to 2021 were selected in this investigation. About missing rates, this review indicated that 60,4% used missing rates $<=$ 30% or did not reveal their missing rates for the experimentation.

The above aspects demonstrate that EvoImp is suitable for missing value treatments in real-world scenarios.

Conclusion and suggestions for future work

The data analyses conducted in real-world datasets make it clear that there is a critical need to handle missing values in multi-label classification domain. The ubiquitous presence of MVs and the fact that most of the techniques employed only work or ensure good performance when applied to datasets with complete cases underlines the need to tackle this problem. Data imputation methods have emerged as an alternative solution, searching for plausible values to fill the missing ones.

Therefore, we proposed in this study the EvoImp, an imputation method based on genetic algorithms for the optimization of multiple imputations for missing data applied to multi-label learning. For validation, the method was submitted to an extensive experimental benchmarking process with various multi-label datasets and compared with other state-of-the-art imputation methods. Six missing value rates are applied to the datasets to simulate the MCAR mechanism. The results were analyzed using five classifiers: Binary Relevance, Hierarchy of Multi-label Classifier, Multi-Label k-Nearest Neighbors, Classifier Chains, and Ensembles of Classifier Chains. Three well-known evaluation measures were adopted to assess the experiments: Exact Match, Accuracy, and Hamming-loss.

EvoImp achieved exceptional results in all the scenarios evaluated, being quantitatively superior to the others. These outstanding results make it possible to conclude that the proposed method is suitable for application in real-world scenarios. In addition to a novel approach for dealing with MV in multi-label classification, the present works contribute to the body of knowledge by: i) assessing the impact of missing data on multi-label classification to improve classification robustness; ii) providing an extensive experimental comparison of many state-of-the-art data imputation algorithms, multi-label machine learning classifiers, and performance measures; iii) making source codes and experiments results in a GitHub repository.

In future work, we want to evaluate other missingness mechanisms apart from MCAR and adjust the method for handling high rates of missing data ($>$ 30%). Experiments could also be

performed to make the EvoImp learn its parameters (AutoML). Finally, we would like to investigate the Influence of Cardinality and Density Characteristics on Multi-Label Learning with missing values.

Supporting information

S1 Table. Abbreviations.
(PDF)

Author Contributions

Conceptualization: Antonio Fernando Lavareda Jacob Junior, Adamo Lima de Santana, Fabio Manoel Franca Lobato.

Data curation: Fabio Manoel Franca Lobato.

Formal analysis: Fabio Manoel Franca Lobato.

Investigation: Antonio Fernando Lavareda Jacob Junior.

Methodology: Fabio Manoel Franca Lobato.

Software: Fabricio Almeida do Carmo.

Supervision: Ewaldo Eder Carvalho Santana, Fabio Manoel Franca Lobato.

Validation: Antonio Fernando Lavareda Jacob Junior, Fabio Manoel Franca Lobato.

Visualization: Antonio Fernando Lavareda Jacob Junior.

Writing – original draft: Antonio Fernando Lavareda Jacob Junior.

Writing – review & editing: Antonio Fernando Lavareda Jacob Junior, Fabricio Almeida do Carmo, Adamo Lima de Santana, Ewaldo Eder Carvalho Santana, Fabio Manoel Franca Lobato.

References

1. Heymans MW, Twisk JW. Handling missing data in clinical research. Journal of clinical epidemiology. 2022 Nov 1; 151:185. https://doi.org/10.1016/j.jclinepi.2022.08.016 PMID: 36150546
2. Honaker J, King G. What to Do about Missing Values in Time-Series Cross-Section Data. American Journal of Political Science. 2010 Apr 54;2:561–581 https://doi.org/10.1111/j.1540-5907.2010.00447.x
3. Tsai CF, Li ML, Lin WC. A class center based approach for missing value imputation. Knowledge-Based Systems. 2018 Jul 151:124–35. https://doi.org/10.1016/j.knosys.2018.03.026
4. Lin WC, Tsai CF. Missing value imputation: a review and analysis of the literature (2006–2017). Artificial Intelligence Review. 2020 Feb; 53:1487–509. https://doi.org/10.1007/s10462-019-09709-4
5. Garciarena U, Santana R. An extensive analysis of the interaction between missing data types, imputation methods, and supervised classifiers. Expert Systems with Applications. 2017 Dec 15; 89:52–65. https://doi.org/10.1016/j.eswa.2017.07.026
6. Adhikari D, Jiang W, Zhan J, He Z, Rawat DB, Aickelin U, et al. A comprehensive survey on imputation of missing data in internet of things. ACM Computing Surveys. 2022 Dec 15; 55(7):1–38. https://doi.org/10.1145/3533381
7. Luengo J, García S, Herrera F. On the choice of the best imputation methods for missing values considering three groups of classification methods. Knowl Inf Syst. 2012; 32(1):77–108. https://doi.org/10.1007/s10115-011-0424-2
8. Emmanuel T, Maupong T, Mpoeleng D, Semong T, Mphago B, Tabona O. A survey on missing data in machine learning. Journal of Big Data. 2021 Dec; 8(1):1–37. https://doi.org/10.1186/s40537-021-00516-9 PMID: 34722113
9. McMahon P, Zhang T, Dwight RA. Approaches to dealing with missing data in railway asset management. IEEE Access. 2020 Mar 6; 8:48177–94. https://doi.org/10.1109/ACCESS.2020.2978902

10. Ren L, Wang T, Seklouli AS, Zhang H, Bouras A. A review on missing values for main challenges and methods. Information Systems. 2023 Oct 119. https://doi.org/10.1016/j.is.2023.102268
11. Farhangfar A, Kurgan L, Dy J. Impact of imputation of missing values on classification error for discrete data. Pattern Recognition. 2008 Dec 1; 41(12):3692–705. https://doi.org/10.1016/j.patcog.2008.05.019
12. Rubin DB. An overview of multiple imputation. In: Proceedings of the survey research methods section of the American statistical association 1988 Aug (Vol. 79, p. 84). Princeton, NJ, USA: Citeseer.
13. Li P, Stuart EA, Allison DB. Multiple imputation: a flexible tool for handling missing data. Jama. 2015 Nov 10; 314(18):1966–7. https://doi.org/10.1001/jama.2015.15281 PMID: 26547468
14. Rubin DB. Multiple imputation for nonresponse in surveys. John Wiley & Sons; 2004 Jun 9.
15. Lobato FMF. Evolutionary strategies to optimize the treatment of missing data by multiple imputation data (*in Portuguese*). PhD Thesis, Federal University of Pará, 2016.
16. Nunes LN, Kluck MM, Fachel JMG. Use of multiple imputation for missing data: a simulation using epidemiological data (*in Portuguese*). Cad Saúde Pública [online]. 2009; 25(2):268–278. https://doi.org/10.1590/S0102-311X2009000200005 PMID: 19219234
17. Chiu PC, Selamat A, Krejcar O, Kuok KK, Bujang SD, Fujita H. Missing Value Imputation Designs and Methods of Nature-Inspired Metaheuristic Techniques: A Systematic Review. IEEE Access. 2022 May 9.(pp. 61544–61566). https://doi.org/10.1109/ACCESS.2022.3172319
18. Holland JH. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press; 1992 Apr 29.
19. Garcia JCF, Kalenatic D, Bello CAL Missing data imputation in multivariate data by evolutionary algorithms. Comput Hum Behav. 2011. 27:1468–1474 https://doi.org/10.1016/j.chb.2010.06.026
20. Provost F, Saar-Tsechanski M. Handling Missing Values when Applying Classification Models. Journal of Machine Learning Research. 2007; 8.
21. Read J, Pfahringer B, Holmes G, Frank E. Classifier chains for multi-label classification. Machine learning. 2011 Dec; 85:333–59. https://doi.org/10.1007/s10994-011-5256-5
22. Ghani MU, Rafi M, Tahir MA. Discriminative adaptive sets for multi-label classification. IEEE Access. 2020 Dec 1; 8:227579–95. https://doi.org/10.1109/ACCESS.2020.3041763
23. Gonçalves EC, Freitas AA, Plastino A. A survey of genetic algorithms for multi-label classification. In: 2018 IEEE Congress on Evolutionary Computation (CEC) 2018 Jul 8 (pp. 1-8). IEEE.
24. Nguyen TT, Nguyen TT, Luong AV, Nguyen QV, Liew AW, Stantic B. Multi-label classification via label correlation and first order feature dependance in a data stream. Pattern recognition. 2019 Jun 1; 90:35–51. https://doi.org/10.1016/j.patcog.2019.01.007
25. de Sá AG, Pimenta CG, Pappa GL, Freitas AA. A robust experimental evaluation of automated multi-label classification methods. In: Proceedings of the 2020 Genetic and Evolutionary Computation Conference 2020 Jun 25 (pp. 175-183).
26. Venkatesan R, Er MJ. Multi-label classification method based on extreme learning machines. In: 2014 13th International Conference on Control Automation Robotics & Vision (ICARCV) 2014 Dec 10 (pp. 619-624). IEEE.
27. Liu W, Wang H, Shen X, Tsang IW. The emerging trends of multi-label learning. IEEE transactions on pattern analysis and machine intelligence. 2021 Oct 12; 44(11):7955–74. https://doi.org/10.1109/TPAMI.2021.3119334
28. Tsoumakas G, Katakis I, Vlahavas I. Random k-labelsets for multilabel classification. IEEE transactions on knowledge and data engineering. 2010 Sep 9; 23(7):1079–89. https://doi.org/10.1109/TKDE.2010.164
29. Tang, Lei and Rajan, Suju and Narayanan, Vijay K. Large scale multi-label classification via metalabeler. In Proceedings of the 18th international conference on World wide web, pages 211–220, 2009.
30. Qian K, Min XY, Cheng Y, Song G, Min F. Self-dependence multi-label learning with double k for missing labels. Artificial Intelligence Review. 2022 Oct 23:1–38.
31. Sun L, Yin T, Ding W, Qian Y, Xu J. Feature selection with missing labels using multilabel fuzzy neighborhood rough sets and maximum relevance minimum redundancy. IEEE Transactions on Fuzzy Systems. 2021 Jan 22; 30(5):1197–211. https://doi.org/10.1109/TFUZZ.2021.3053844
32. Gibaja E, Ventura S. A tutorial on multilabel learning. ACM Computing Surveys (CSUR). 2015 Apr 16; 47(3):1–38. https://doi.org/10.1145/2716262
33. Pereira RB, Plastino A, Zadrozny B, Merschmann LH. Correlation analysis of performance measures for multi-label classification. Information Processing & Management. 2018 May 1; 54(3):359–69. https://doi.org/10.1016/j.ipm.2018.01.002
34. Zheng X, Li P, Chu Z, Hu X. A survey on multi-label data stream classification. IEEE Access. 2019 Dec 24; 8:1249–75. https://doi.org/10.1109/ACCESS.2019.2962059

35. Wang C, Lin Y, Liu J. Feature selection for multi-label learning with missing labels. Applied Intelligence. 2019 Aug 15; 49:3027–42. https://doi.org/10.1007/s10489-019-01431-6

36. Cheng Y, Song F, Qian K. Missing multi-label learning with non-equilibrium based on two-level autoencoder. Applied Intelligence. 2021 Oct 1:1–9.

37. Tran CT, Zhang M, Andreae P. Multiple imputation for missing data using genetic programming. In: Proceedings of the 2015 annual conference on genetic and evolutionary computation 2015 Jul 11 (pp. 583-590).

38. Shahzad W, Rehman Q, Ahmed E. Missing Data Imputation using Genetic Algorithm for Supervised Learning. Int J Adv Comput Sci Appl. 2017; 8.

39. Lobato F, Sales C, Araujo I, Tadaiesky V, Dias L, Ramos L, et al. Multi-objective genetic algorithm for missing data imputation. Pattern Recognition Letters. 2015 Dec 15; 68:126–31. https://doi.org/10.1016/j.patrec.2015.08.023

40. Mirjalili S. Genetic Algorithm. In: Evolutionary Algorithms and Neural Networks. Studies in Computational Intelligence. 2019; 780. Springer.

41. Karafotias, Giorgos, Mark Hoogendoorn, and AE Eiben. Evaluating reward definitions for parameter control. In Proceedings of the 18th European Conference on Applications of Evolutionary Computation (EvoApplications 2015), Copenhagen, Denmark, April 8-10, 2015, pp. 667-680. Springer, 2015.

42. Reynoso-Meza, Gilberto, Javier Sanchis, Xavier Blasco, and Juan M Herrero. Hybrid DE algorithm with adaptive crossover operator for solving real-world numerical optimization problems. In Proceedings of the 2011 IEEE Congress of Evolutionary Computation (CEC), pp. 1551-1556. IEEE, 2011.

43. Semenkin E, Semenkina M. Self-configuring genetic algorithm with modified uniform crossover operator. In: Tan Y, Shi Y, Ji Z, editors. Advances in Swarm Intelligence. ICSI 2012. Lecture Notes in Computer Science. 2012; 7331. Springer. pp. 414–421.

44. Lobato FMF, Tadaiesky VW, Araújo IM,de Santana ÁL. An Evolutionary Missing Data Imputation Method for Pattern Classification. In: Proc. Genet Evol Comput Conf—GECCO. 2015.

45. Gonçalves EC, Plastino A, Freitas AA. A genetic algorithm for optimizing the label ordering in multi-label classifier chains. In: Proc. Int. Conf. Tools with Artif. Intell. ICTAI. 2013. pp. 469–476.

46. González J, Ortega J, Escobar JJ, Damas M. A lexicographic cooperative co-evolutionary approach for feature selection. Neurocomputing. 2021; 463:59–76. https://doi.org/10.1016/j.neucom.2021.08.003

47. González J, Ortega J, Damas M, Martín-Smith P. Many-objective cooperative co-evolutionary feature selection: A lexicographic approach. In: Rojas I, Joya G, Catalá A, editors. Advances in Computational Intelligence, IWANN 2019. Lecture Notes in Computer Science. 2019; 11507. Springer. pp. 463–474.

48. Esmaeili A, Behdin K, Fakharian MA, Marvasti F. Transductive multi-label learning from missing data using smoothed rank function. Pattern Anal Applic. 2020; 23:1225–1233. https://doi.org/10.1007/s10044-020-00869-6

49. Santos MS, Pereira RC, Costa AF, Soares JP, Santos J, Abreu PH. Generating Synthetic Missing Data: A Review by Missing Mechanism. IEEE Access. 2019; 7:11651–11667. https://doi.org/10.1109/ACCESS.2019.2891360

50. Tsoumakas G, Katakis I, Vlahavas I. Effective and efficient multilabel classification in domains with large number of labels. In: Proc. ECML/PKDD 2008 Work. Min. Multidimens. Data. 2008. pp. 30–44.

51. Tsoumakas G, Spyromitros-Xioufis E, Vilcek J, Vlahavas I. MULAN: A Java library for multi-label learning. J Mach Learn Res. 2011; 12:2411–2414.

52. Frank E, Hall MA, Witten IH. The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques". Morgan Kaufmann, Fourth Edition. 2016.

53. Triguero I, González S, Moyano JM, García S, Alcalá-Fdez J, Luengo J, et al. KEEL 3.0: An Open Source Software for Multi-Stage Analysis in Data Mining. Int J Comput Intell Syst. 2017; 10:1238–1249. https://doi.org/10.2991/ijcis.10.1.82

54. Schmitt P, Mandel J, Guedj M. A comparison of six methods for missing data imputation. Journal of Biometrics & Biostatistics. 2015 Jan 1; 6(1):1.