Universidade Federal de Campina Grande

Centro de Engenharia Elétrica e Informática

Coordenação de Pós-Graduação em Engenharia Elétrica

Andressa Carvalho Melo da Silveira

Análise Formal de Sistemas Baseados em Aprendizado de Máquina Usando Redes de Petri Coloridas

Campina Grande, Paraíba, Brasil
2025

Universidade Federal de Campina Grande

Centro de Engenharia Elétrica e Informática

Coordenação de Pós-Graduação em Engenharia Elétrica

Análise Formal de Sistemas Baseados em Aprendizado de Máquina Usando Redes de Petri Coloridas

Andressa Carvalho Melo da Silveira

Tese de Doutorado submetida ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para obtenção do grau de Doutora em Engenharia Elétrica

Área de Concentração: Processamento da Informação

Linha de Pesquisa: Engenharia de Computação

Angelo Perkusich, D.Sc

Álvaro Alvares de Carvalho César Sobrinho, D.Sc

(Orientadores)

Campina Grande, Paraíba, Brasil

Análise Formal de Sistemas Baseados em Aprendizado de Máquina Usando Redes de Petri Coloridas

ANDRESSA CARVALHO MELO DA SILVEIRA

TESE APROVADA EM 05/12/2024

ANGELO PERKUSICH, Dr, UFCG
Orientador(a)

ÁLVARO ALVARES DE CARVALHO CÉSAR SOBRINHO, Dr., UFAPE
Orientador(a)

PÉRICLES REZENDE BARROS, Ph.D., UFCG
Examinador(a)

EISENHAWER DE MOURA FERNANDES, D.Sc., UFCG
Examinador(a)

JAIDILSON JÓ DA SILVA, D.Sc., UFCG
Examinador(a)

EVANDRO DE BARROS COSTA, D.Sc., UFAL
Examinador(a)

RAFAEL FERREIRA LEITE DE MELLO, Dr, UFRPE
Examinador(a)

CAMPINA GRANDE - PB

S587a Silveira, Andressa Carvalho Melo da.

Análise formal de sistemas baseados em apredizado de máquina usando redes de petri coloridas / Andressa Carvalho Melo da Silveira. – Campina Grande, 2024.

157 f. : il. color.

Tese (Doutorado em Engenharia Elétrica) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2024.

"Orientação: Prof. Dr. Angelo Perkusich, Prof. Dr. Álvaro Alvares de Carvalho César Sobrinho".

Referências.

1. Aprendizado de Máquina. 2. Decision Tree. 3. Redes de Petri Colorida. 4. Métodos Formais. I. Perkusich, Angelo. II. César Sobrinho, Álvaro Alvares de Carvalho. III. Título.

CDU 004.8(043)

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECÁRIA MEIRE EMANUELA DA SILVA MELO CRB-15/568

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
POS-GRADUACAO EM ENGENHARIA ELETRICA
Rua Aprigio Veloso, 882, - Bairro Universitario, Campina Grande/PB, CEP 58429-900

REGISTRO DE PRESENÇA E ASSINATURAS

1. ATA DA DEFESA PARA CONCESSÃO DO GRAU DE DOUTOR EM CIÊNCIAS, NO DOMÍNIO DA ENGENHARIA ELÉTRICA, REALIZADA EM 05 DE DEZEMBRO DE 2024
(Nº 389)

CANDIDATO(A): **ANDRESSA CARVALHO MELO DA SILVEIRA.** COMISSÃO EXAMINADORA: PÉRICLES REZENDE BARROS, Ph.D., UFCG - Presidente da Comissão e Examinador Interno, ANGELO PERKUSICH, D.Sc., UFCG -Orientador, ÁLVARO ALVARES DE CARVALHO CÉSAR SOBRINHO , D.Sc., UFAPE - Orientador, EISENHAWER DE MOURA FERNANDES, D.SC., UFCG - Examinador Interno, JAIDILSON JÓ DA SILVA, D.SC., UFCG - Examinador Externo, EVANDRO DE BARROS COSTA , D.SC., UFAL - Examinador Externo, RAFAEL FERREIRA LEITE DE MELLO, D.SC., UFRPE - Examinador Externo. TÍTULO DA TESE: Análise Formal de Sistemas Baseados em Aprendizado de Máquina Usando Redes de Petri Coloridas. ÁREA DE CONCENTRAÇÃO: Processamento da Informação. HORA DE INÍCIO: **08h00** – LOCAL: **Sala Virtual, conforme Art. 5º da PORTARIA SEI Nº 01/PRPG/UFCG/GPR, DE 09 DE MAIO DE 2022.** Em sessão pública, após exposição de cerca de 45 minutos, o(a) candidato(a) foi arguido(a) oralmente pelos membros da Comissão Examinadora, tendo demonstrado suficiência de conhecimento e capacidade de sistematização, no tema de sua tese, obtendo conceito EM EXIGÊNCIA, o candidato terá até 90 (noventa) dias, conforme decisão da Comissão, para providenciar as alterações exigidas, conforme lista estabelecida, constante nos documentos de AVALIAÇÃO DE TESE DE DOUTORADO enviados pelos membros da Comissão Examinadora. Face EM EXIGÊNCIA, declara o presidente da Comissão, achar-se o examinado, legalmente habilitado a receber o Grau de Doutor em Ciências, no domínio da Engenharia Elétrica, após atender as exigências da banca. Atendimento este que deverá ser atestado pelos orientadores, quando caberá à Universidade Federal de Campina Grande, como de direito, providenciar a expedição do Diploma, a que o(a) mesmo(a) fará jus. Na forma regulamentar, foi lavrada a presente ata, que é assinada por mim, Leandro Ferreira de Lima, e os membros da Comissão Examinadora. Campina Grande, 05 de Dezembro de 2024.

LEANDRO FERREIRA DE LIMA

Secretário

PÉRICLES REZENDE BARROS, Ph.D., UFCG

Presidente da Comissão e Examinador Interno

ANGELO PERKUSICH, D.Sc., UFCG

Orientador

ÁLVARO ALVARES DE CARVALHO CÉSAR SOBRINHO, D.Sc., UFAPE
Orientador

EISENHAWER DE MOURA FERNANDES, D.SC., UFCG
Examinador Interno

JAIDILSON JÓ DA SILVA, D.SC., UFCG
Examinador Externo

EVANDRO DE BARROS COSTA , D.SC., UFAL

Examinador Externo

RAFAEL FERREIRA LEITE DE MELLO, D.SC., UFRPE
Examinador Externo

ANDRESSA CARVALHO MELO DA SILVEIRA

Candidata

2 - APROVAÇÃO

2.1. Segue a presente Ata de Defesa de Tese de Doutorado da candidato **ANDRESSA CARVALHO MELO DA SILVEIRA,** assinada eletronicamente pela Comissão Examinadora acima identificada.

2.2. No caso de examinadores externos que não possuam credenciamento de usuário externo ativo no SEI, para igual assinatura eletrônica, os examinadores internos signatários **certificam** que os examinadores externos acima identificados participaram da defesa da tese e tomaram conhecimento do teor deste documento.

Documento assinado eletronicamente por **LEANDRO FERREIRA DE LIMA**, **SECRETÁRIO (A)**, em 06/12/2024, às 15:35, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **JAIDILSON JO DA SILVA**, **PROFESSOR(A) DO MAGISTERIO SUPERIOR**, em 06/12/2024, às 16:01, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **PERICLES REZENDE BARROS**, **PROFESSOR 3 GRAU**, em 06/12/2024, às 16:13, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **EISENHAWER DE MOURA FERNANDES**, **PROFESSOR(A) DO MAGISTERIO SUPERIOR**, em 07/12/2024, às 11:15, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **Álvaro Alvares de Carvalho César Sobrinho**, **Usuário Externo**, em 09/12/2024, às 08:44, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **ANGELO PERKUSICH**, **PROFESSOR(A) DO MAGISTERIO SUPERIOR**, em 09/12/2024, às 10:14, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **Evandro de Barros Costa**, **Usuário Externo**, em 10/12/2024, às 10:39, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **ANDRESSA CARVALHO MELO DA SILVEIRA QUEIROZ**, **Usuário Externo**, em 03/02/2025, às 19:44, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **Rafael Ferreira Leite de Mello**, **Usuário Externo**, em 06/02/2025, às 11:28, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

A autenticidade deste documento pode ser conferida no site https://sei.ufcg.edu.br/autenticidade, informando o código verificador **5102728** e o código CRC **A2837CFA**.

Referência: Processo nº 23096.088977/2024-18 SEI nº 5102728

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CNPJ nº 05.055.128/0001-76
POS-GRADUACAO EM ENGENHARIA ELETRICA
Rua Aprigio Veloso, 882, - Bairro Universitario, Campina Grande/PB, CEP 58429-900

DECLARAÇÃO

Processo nº 23096.088977/2024-18

DECLARAMOS para fins de comprovação que, os Professores PÉRICLES REZENDE BARROS, Ph.D., UFCG - Presidente da Comissão e Examinador Interno, ANGELO PERKUSICH, D.Sc., UFCG - Orientador, ÁLVARO ALVARES DE CARVALHO CÉSAR SOBRINHO , D.Sc., UFAPE - Orientador, EISENHAWER DE MOURA FERNANDES, D.SC., UFCG - Examinador Interno, JAIDILSON JÓ DA SILVA, D.SC., UFCG - Examinador Externo, EVANDRO DE BARROS COSTA , D.SC., UFAL - Examinador Externo, RAFAEL FERREIRA LEITE DE MELLO, D.SC., UFRPE - Examinador Externo, participaram da Banca de Defesa Final da Tese de Doutorado, do Programa de Pós- Graduação em Engenharia Elétrica da UFCG, intitulada TÍTULO DA TESE: **Análise Formal de Sistemas Baseados em Aprendizado de Máquina Usando Redes de Petri Coloridas**, de autoria da doutoranda **ANDRESSA CARVALHO MELO DA SILVEIRA,** no dia 05 de dezembro de 2024.

Documento assinado eletronicamente por **ALEXANDRE JEAN RENE SERRES**, **COORDENADOR(A)**, em 09/12/2024, às 09:22, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

A autenticidade deste documento pode ser conferida no site https://sei.ufcg.edu.br/autenticidade, informando o código verificador **5102850** e o código CRC **73086902**.

Referência: Processo nº 23096.088977/2024-18 SEI nº 5102850

Resumo

O Aprendizado de Máquina (AM) tem sido amplamente aplicado em áreas críticas como saúde, manufatura e transporte. No entanto, sua integração em sistemas críticos exige maior explicabilidade e acurácia. Modelos como árvores de decisão (*Decision Tree - DT*) e florestas aleatórias (*Random Forest - RF*) podem gerar regras redundantes, dificultando a interpretação e comprometendo a transparência. As DTs aumentam em profundidade e número de nós ao capturar padrões, o que pode dificultar a interpretação, comprometendo a transparência e a aplicabilidade em sistemas críticos, especialmente na saúde. Nesta tese, é apresentado um método baseado em redes de Petri coloridas (*Coloured Petri Nets - CPN*) que visa melhorar a explicabilidade de modelos DT e RF. O método, denominado RuleXtract/CPN, automatiza a extração, análise e ajuste de regras de decisão, além de permitir que essas etapas sejam realizadas por usuários sem expertise em CPN. O método desenvolvido consiste em transformar modelos DT e RF em modelos de CPN. Por meio de simulações, as regras de decisão são analisadas e ajustadas, eliminando redundâncias e identificando regras específicas ou incorretas que geram classificações enganosas. A implementação do método foi realizada com tecnologias web integradas ao arcabouço Access/CPN, de modo que os usuários não precisem ter experiência em CPN para gerar e simular modelos, executando-os em segundo plano. Experimentos foram conduzidos com seis conjuntos de dados relacionados à COVID-19 e cinco de Influenza. Os resultados mostram uma redução significativa no número de regras de decisão: no conjunto balanceado, as regras foram reduzidas de 882 para 688, enquanto, no conjunto desbalanceado, a redução foi de 876 para 687. A eliminação de regras redundantes reduziu a complexidade dos modelos, facilitando a validação por especialistas antes da adoção das regras em um sistema de suporte à decisão (*Decision Support System - DSS*). Os achados destacam a relevância do método para aumentar a confiança e a explicabilidade de modelos de AM aplicados a sistemas críticos. A metodologia desenvolvida apresenta potencial para pesquisas futuras, incluindo sua escalabilidade e aplicação a outros algoritmos de AM.

Abstract

Machine Learning (ML) has been widely applied in critical areas such as healthcare, manufacturing, and transportation. However, its integration into critical systems requires greater explainability and accuracy. Models like Decision Trees (DT) and Random Forests (RF) can generate redundant rules, making interpretation difficult and compromising transparency. DTs increase in depth and number of nodes as they capture patterns, which can hinder interpretation, affecting transparency and applicability in critical systems, particularly in healthcare. This thesis presents a method based on Coloured Petri Nets (CPN) aimed at improving the explainability of DT and RF models. The method, named RuleXtract/CPN, automates the extraction, analysis, and adjustment of decision rules, allowing these steps to be performed by users without expertise in CPN. The developed method consists of transforming DT and RF models into CPN models. Through simulations, the decision rules are analyzed and adjusted, eliminating redundancies and identifying specific or incorrect rules that produce misleading classifications. The method was implemented using web technologies integrated with the Access/CPN framework, so users do not need CPN expertise to generate and simulate models, executing them in the background. Experiments were conducted with six COVID-19-related datasets and five related to Influenza. The results show a significant reduction in the number of decision rules: in the balanced dataset, the rules were reduced from 882 to 688, while in the imbalanced dataset, the reduction was from 876 to 687. The elimination of redundant rules reduced the complexity of the models, making it easier for experts to validate them before adopting the rules in a Decision Support System (DSS). The findings highlight the relevance of the method in increasing trust and explainability of ML models applied to critical systems. The developed methodology presents potential for future research, including its scalability and application to other ML algorithms.

Conteúdo

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Muitos sistemas críticos dependem de técnicas de Aprendizado de Máquina (AM), impulsionando avanços em várias indústrias, como a de saúde. Nessas áreas, a AM possibilita avanços importantes, como o diagnóstico precoce e a melhoria de tratamentos, por meio de sistemas de apoio à decisão (*Decision Support Systems - DSS*) [100]. Esses sistemas são importantes porque fatores como falta de conhecimento e julgamentos incorretos podem impactar negativamente o processo de tomada de decisão [68]. A Figura 1.1 apresenta os principais pilares do uso de AM na área da saúde, demonstrando de que forma essa tecnologia contribui para a melhoria da qualidade dos serviços médicos. Dentre esses pilares, destacam-se [50]:

- previsão de surtos (*Outbreak Prediction*), que viabiliza a antecipação de epidemias e a implementação de medidas preventivas;
- descoberta e fabricação de medicamentos (*Drug Discovery and Manufacturing*), acelerando o desenvolvimento de novos tratamentos;
- diagnóstico médico (*Medical Imaging Diagnosis*), com algoritmos que ampliam a precisão na interpretação de exames;
- modificação comportamental (*Behavioral Modification*), promovendo mudanças de hábitos para melhoria da saúde;
- radioterapia avançada (*Better Radiotherapy*), que aperfeiçoa os tratamentos oncológi-

cos;

- coleta de dados colaborativa (*Crowdsourced Data Collection*), integrando dados de diferentes fontes para análises mais completas;
- ensaios clínicos e pesquisa (*Clinical Trial and Research*), que otimizam os estudos científicos; e
- registros inteligentes de saúde (*Smart Health Records*), facilitando a gestão e o acesso aos dados dos pacientes.

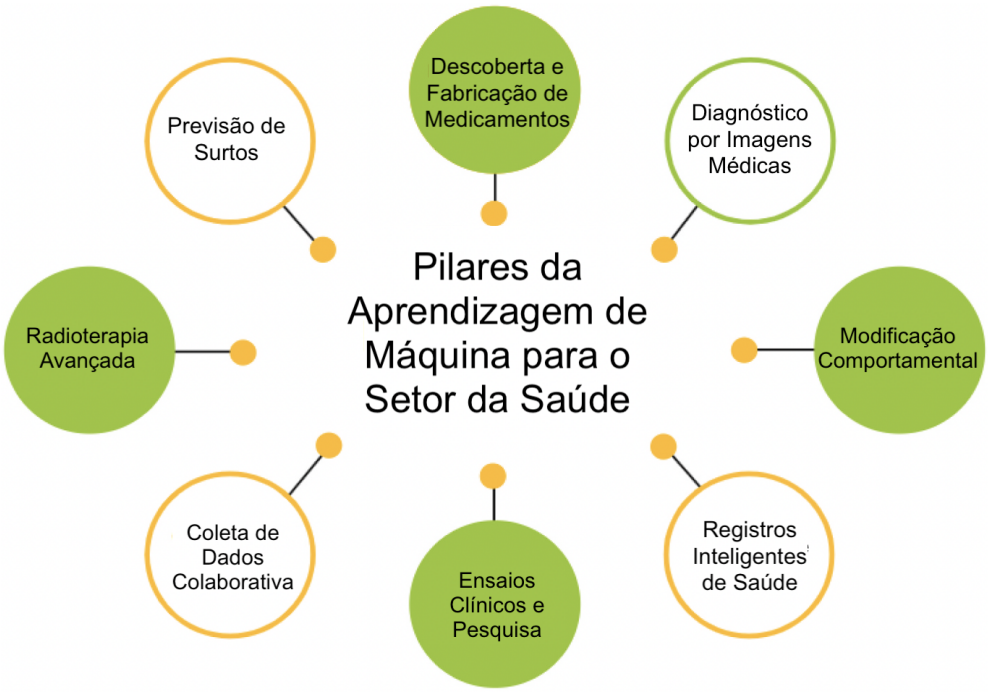

Figura 1.1: Pilares de AM no cuidado à saúde [50].

Entre os pilares destacados, o estudo de [50] enfatiza o diagnóstico por imagens médicas (*Medical Imaging Diagnosis*) como uma aplicação de grande relevância. No entanto, o diagnóstico vai além da análise de imagens, abrangendo também a interpretação de exames laboratoriais e outros dados clínicos. Essa visão ampliada demonstra o potencial da AM em integrar diferentes tipos de informações para oferecer suporte abrangente e preciso na identificação de condições de saúde.

Nesse contexto, este trabalho foca especificamente no diagnóstico, reconhecido como um dos pilares fundamentais da aplicação da AM na saúde. A escolha do diagnóstico como tema central reflete sua importância crítica para o setor, considerando os impactos diretos na segurança do paciente, na eficácia do tratamento e na redução de erros clínicos.

Embora a relevância da AM se estenda além do setor de saúde, sua importância se destaca especialmente nos DSS na área médica. Esses sistemas utilizam técnicas de AM para analisar grandes volumes de dados de saúde e auxiliar profissionais em processos complexos de tomada de decisão. Por meio da combinação de dados, modelos analíticos e interfaces intuitivas, os DSS resolvem tanto problemas estruturados quanto não estruturados. Na prática, têm desempenhado um papel essencial em diagnósticos, tratamentos e prognósticos, baseando-se em padrões extraídos de dados clínicos [13].

A necessidade de DSS é evidenciada pelo fato de que, em muitos casos, a falta de conhecimento ou decisões equivocadas impactam negativamente o processo de tomada de decisão clínica [68]. Estudos demonstram que erros de diagnóstico representam um problema significativo na área da saúde. Por exemplo, uma análise de autópsias revelou que entre 10% e 15% dos casos continham erros de diagnóstico graves [116]. Além disso, em pacientes internados, estudos indicam que entre 6% e 17% dos eventos adversos, situações que prejudicam o paciente, como complicações inesperadas, estão associados a diagnósticos errados [116]. Em um estudo realizado na Holanda, dentre os erros de diagnósticos confirmados, verificou-se que quase todos (96,3%) os erros estavam ligados a falhas humanas, como conhecimento inadequado ou à aplicação incorreta de informações médicas [139]. Já no atendimento ambulatorial, cerca de 5% dos pacientes recebem diagnósticos incorretos devido a oportunidades perdidas para interpretar corretamente os dados disponíveis, como resultados de exames e sintomas relatados pelos pacientes [109]. Em estudos de casos médicos, Graber et al. [38] destacaram que a maioria dos erros de diagnóstico resulta de limitações cognitivas dos profissionais de saúde ou falhas nos sistemas de suporte existentes. Além disso, estimativas indicam que 25% dos exames realizados após a morte dos pacientes revelam erros críticos relacionados ao diagnóstico principal ou à causa do óbito [107].

Essas evidências reforçam a necessidade de DSS baseados em AM, que podem melhorar

não apenas a precisão no diagnóstico, mas também fornecer explicações claras e compreensíveis de suas recomendações. Ao estruturar e apresentar informações fundamentadas, esses sistemas ajudam os profissionais de saúde a reduzir erros cognitivos e tomar decisões mais seguras e eficazes, beneficiando diretamente os pacientes.

A Figura 1.2 ilustra as principais vantagens e desvantagens dos DSS aplicados à saúde. Por um lado, os DSS oferecem benefícios claros, como a redução de erros cognitivos, a detecção precoce de condições médicas, o auxílio aos profissionais na tomada de decisão e a identificação de padrões complexos. Por outro lado, enfrentam desafios significativos, como os riscos associados a falhas, consequências críticas em casos de erro, limitações na explicabilidade das recomendações e dificuldades na implementação e manutenção.

Figura 1.2: Vantagens e desvantagens dos Sistemas de Apoio à Decisão na saúde.

A explicabilidade é um dos principais desafios, já que as razões por trás de uma saída ou decisão nem sempre são claras, sobretudo em modelos complexos, como os de aprendizado profundo. Nesse cenário, a explicabilidade torna-se crucial para DSS aplicados à saúde, onde decisões precisam ser justificadas, auditadas e confiáveis [4]. Outro desafio é a qualidade e representatividade dos dados usados para treinar e validar os modelos. Dados tendenciosos ou inadequados comprometem a capacidade de generalização dos sistemas, levando a resultados não confiáveis [55].

Além disso, regras duplicadas ou conflitantes podem aumentar a complexidade do modelo,

dificultando sua interpretação e validação. Esse problema impacta diretamente a confiabilidade do modelo, pois a presença de regras redundantes pode gerar inconsistências na tomada de decisão. Em sistemas críticos, como na área da saúde, essa falta de clareza pode comprometer a adoção do modelo, tornando essencial a remoção de regras desnecessárias para garantir decisões mais transparentes e justificáveis.

Isso ressalta a necessidade de avanços contínuos no desenvolvimento de DSS que combinem alta precisão com confiabilidade e explicabilidade, facilitando sua adoção prática e maximizando o impacto positivo no setor de saúde. Para superar essas questões, é fundamental que as regras de decisão sejam revisadas por especialistas antes de serem implementadas. Essa validação por especialistas assegura que as regras sejam relevantes, consistentes com as melhores práticas clínicas e alinhadas às necessidades reais do sistema [122]. Além disso, a revisão aumenta a aceitação e a confiança dos profissionais no uso do sistema, promovendo sua adoção em ambientes críticos de maneira mais segura e eficaz.

Neste trabalho, propõe-se um método baseado em redes de Petri coloridas (Coloured Petri Nets - CPN) para fortalecer a confiabilidade e a explicabilidade de DSS baseados em AM. A complexidade crescente de modelos como árvores de decisão (*Decision Trees - DT*) e florestas aleatórias (*Random Forests - RF*) pode gerar regras redundantes ou conflitantes, dificultando sua interpretação e validação. O método desenvolvido, denominado RuleXtract/CPN, automatiza a extração, análise e ajuste das regras de decisão, utilizando simulação para eliminar redundâncias e identificar regras problemáticas. Dessa forma, busca-se aprimorar a transparência dos modelos e facilitar sua validação por especialistas, tornando os DSS mais robustos e confiáveis, especialmente em aplicações críticas, como a área da saúde.

CPN emergem como uma solução para lidar com essas questões. Esses métodos permitem a modelagem, análise e validação detalhada das regras de decisão antes de sua implementação em DSS, garantindo que inconsistências sejam eliminadas e classificações enganosas sejam ajustadas. Ao possibilitar a rastreabilidade das decisões, as CPN tornam o processo mais claro e confiável. A combinação de métodos formais com AM não só reforça a segurança, mas também potencializa a eficácia desses sistemas, especialmente em setores críticos como a saúde.

1.1 Justificativa

DSS clínica, que utilizam técnicas de AM, têm se tornado cada vez mais comuns na área da saúde, desempenhando um papel importante na redução de erros durante o processo de tomada de decisão clínica. Esses sistemas auxiliam os profissionais a diagnosticar doenças e definir tratamentos mais adequados. Isso é possível porque os modelos de AM analisam padrões nos dados, o que torna os diagnósticos mais ágeis e precisos. Modelos baseados em regras de decisão, como DT e RF, são amplamente empregados em DSS clínicos [105] [37]. Esses modelos destacam-se por sua estrutura explicável que permite que especialistas revisem as decisões e confiem nas recomendações oferecidas [98].

Em contextos críticos, como a saúde, a literatura ressalta as vantagens dos modelos explicáveis, como DT e RF, em comparação com modelos de aprendizado profundo, conhecidos por sua alta complexidade e falta de transparência. Embora os modelos de aprendizado profundo sejam frequentemente elogiados por sua acurácia, sua natureza de caixa-preta pode torna-los inadequados para aplicações de alto risco, como diagnósticos médicos. Rudin [98] argumenta que, em situações onde vidas humanas estão em jogo, a prioridade deve ser dada a modelos explicáveis, pois eles permitem que especialistas compreendam e validem as decisões automatizadas, minimizando os riscos associados à falta de transparência.

Lipton [67] complementa essa visão, destacando que a ausência de explicabilidade em modelos caixa-preta aumenta os riscos em aplicações sensíveis, como diagnósticos médicos, ao dificultar a compreensão e a validação das decisões. Modelos explicáveis, por outro lado, oferecem maior confiança e clareza, permitindo que profissionais intervenham de forma eficaz quando necessário, evitando o esforço excessivo de justificar decisões oriundas de sistemas complexos e pouco transparentes [81].

Apesar dessas vantagens, modelos de DT e RF apresentam desafios como a duplicação de regras de decisão [85] [83] e *overfitting* [121], que comprometem sua capacidade de generalização, levando a possíveis erros clínicos. Erros de classificação, como falsos negativos, são uma grande preocupação em sistemas de DSS. Esses erros ocorrem quando um paciente com uma condição médica é erroneamente classificado como saudável, o que pode atrasar diagnósticos, levar a tratamentos inadequados e, em casos extremos, resultar em consequên-

cias fatais. Por exemplo, um modelo de AM pode classificar equivocadamente pacientes com COVID-19 ou Influenza como negativos, mesmo em estágios avançados da doença, privando-os de cuidados adequados.

Nos DSS clínicos, a quantidade e a complexidade das regras são fatores fundamentais para garantir a explicabilidade e a eficácia do sistema. Modelos excessivamente complexos dificultam a revisão por especialistas e comprometem a confiança no sistema [62]. Por outro lado, árvores otimizadas oferecem maior transparência, permitindo que médicos e outros profissionais validem as recomendações de forma eficaz, promovendo a aceitação e o uso confiável dos modelos no contexto clínico, em alinhamento com as melhores práticas médicas.

Estudos recentes destacam que a "complexidade da decisão", definida pelo número e pela redundância de regras, pode impactar negativamente a explicabilidade dos modelos. No contexto clínico, essa complexidade não apenas dificulta a validação por especialistas, mas também reduz a eficácia prática dos DSS. Por exemplo, DT otimizadas com métricas de interpretabilidade conseguem manter a acurácia em níveis competitivos, com um custo de apenas 4,2% em relação a modelos mais complexos [54]. Isso demonstra que simplificar as regras e minimizar a profundidade da árvore pode melhorar tanto a eficiência computacional quanto a confiança dos profissionais no sistema.

Alòs et al. (2023) [3] reforçam essa abordagem ao demonstrar que simplificar DT, reduzindo a quantidade e a profundidade das regras, não compromete a precisão e ainda melhora a explicabilidade. Essa estratégia está alinhada ao princípio da *Occam's Razor*, que favorece soluções mais simples quando produzem os mesmos resultados [6]. Na prática, isso resulta na criação de árvores menos complexas e mais fáceis de interpretar, ampliando sua aplicabilidade em cenários críticos, como diagnósticos médicos.

Métodos formais oferecem uma solução promissora para enfrentar os desafios relacionados à complexidade e à explicabilidade. Eles são essenciais para aprimorar a precisão, a confiabilidade, o gerenciamento da complexidade, a explicabilidade, bem como para a validação e verificação de sistemas críticos baseados em AM. Esses métodos se baseiam em uma sólida fundamentação matemática, o que permite aos modeladores melhorar a qualidade por meio

da verificação e validação de sistemas complexos [131]. A modelagem permite representações detalhadas do sistema, ajudando a identificar inconsistências e erros na tomada de decisão no início do processo de desenvolvimento, reduzindo o risco de resultados inesperados [57].

Automatizar a geração de modelos formais pode proporcionar precisão, o que ajuda a prevenir erros de modelagem e elimina a necessidade de conhecimento especializado para usar ferramentas de análise formal em cenários do mundo real. Os modelos formais resultantes podem melhorar o gerenciamento da complexidade no processo de tomada de decisão, que geralmente envolve um conjunto de regras de decisão complexas que podem impactar positivamente ou negativamente os resultados. Essa capacidade de gerenciar a complexidade também pode melhorar a explicabilidade do processo de tomada de decisão, aprimorando a compreensão dos profissionais sobre o raciocínio por trás de recomendações específicas, rastreando as regras e os dados, o que aumenta a confiança.

Por exemplo, os modelos formais podem aprimorar os caminhos clínicos melhorando o gerenciamento de recursos e a análise de sistemas em aplicações de saúde [17]. Outros exemplos que demonstram a relevância dos modelos formais incluem a otimização do desempenho de salas de emergência para reduzir os tempos de espera e melhorar a qualidade do serviço [39] [87] [24], o gerenciamento mais eficaz do fluxo de pacientes e recursos [127] [99], e a melhoria dos protocolos de tratamento [12]. Para sistemas críticos baseados em AM, os modelos formais podem habilitar procedimentos relevantes, como a verificação e a explicabilidade do processo de tomada de decisão [65, 94].

O uso de técnicas como MaxSAT para otimizar DT Puras Mínimas (MPDTs) possibilita a criação de modelos que preservam altos níveis de acurácia, mesmo com conjuntos de regras reduzidos [3]. Árvores menores, de acordo com o estudo, também ajudam a mitigar o problema do *overfitting*, tornando os modelos mais robustos a dados não vistos [3]. Isso evidencia que é possível simplificar os modelos sem comprometer o desempenho, tornando-os ideais para aplicações em áreas críticas.

CPN é particularmente adequado para o gerenciamento da complexidade na tomada de decisões, possibilitando a mineração de processos. *Softwares*, como o CPN Tools ou o CPN

IDE, permitem a análise de desempenho por meio da simulação de modelos para a área de saúde e além [110]. Ao contrário dos *frameworks* ou bibliotecas padrão usados para implementar modelos de DT ou RF, o CPN oferece a vantagem de modelos formais executáveis e uma linguagem de programação funcional durante a análise, o que permite a representação de todos os tipos de dados comumente usados no AM. Conjuntos de cores são importantes para lidar com as diferentes características usadas para resolver problemas de classificação.

O gerenciamento da complexidade é crucial pois, à medida que as regras de decisão aumentam, os modelos de DT e RF podem se tornar menos explicáveis. Relatórios gerados a partir de simulações de modelos CPN podem rastrear todas as decisões internas ao resolver um problema de classificação específico. Além disso, o método desenvolvido nesta tese produz um modelo CPN ajustado que pode ser refinado ainda mais (por exemplo, correção de constantes) com a contribuição de um especialista da área. O uso de *frameworks* ou bibliotecas padrão para AM sem um modelo CPN seria mais limitado, pois um especialista só poderia revisar uma lista estática de regras ou uma representação gráfica de centenas ou milhares de regras. Em contraste, um modelo CPN permite uma análise mais profunda por meio de simulações passo a passo, facilitando a mineração de processos com conceitos como monitores.

1.2 Problemática

O uso de técnicas de AM em DSS, especialmente na área da saúde, é essencial para melhorar a precisão, confiabilidade e explicabilidade dos modelos. Contudo, esses sistemas enfrentam desafios críticos que comprometem sua eficácia e adoção, sobretudo em aplicações sensíveis, como o diagnóstico médico. Falhas nesses modelos podem levar a diagnósticos errados, tratamentos inadequados e, em última instância, a consequências graves para os pacientes.

Modelos baseados em DT e RF, embora sejam mais explicáveis em comparação com métodos mais complexos, como redes neurais profundas, apresentam limitações específicas que afetam sua eficácia. Entre os principais problemas enfrentados estão:

- **regras duplicadas:** duas ou mais regras são consideradas duplicadas quando todos os nós de decisão são idênticos, com exceção de um nó que difere apenas nos operadores

lógicos, resultando em um ramo de decisão redundante. Essa redundância não impacta diretamente a acurácia, mas reduz a clareza e dificulta a revisão por especialistas, comprometendo a explicabilidade e aumentando a complexidade do modelo;

- **regras específicas:** uma regra é considerada específica quando ela está baseada em instâncias particulares do conjunto de treinamento, levando o modelo de AM a memorizar esses dados específicos. Isso resulta na incapacidade do modelo de classificar corretamente novos dados no conjunto de teste. Regras específicas são problemáticas porque limitam a capacidade de generalização do modelo. Essas regras podem ser identificadas analisando o número de *tokens* consumidos por cada transição que a regra representa. Por exemplo, uma regra que classifica apenas uma única instância pode indicar uma regra específica. Um especialista também deve revisar essas regras para determinar se precisam ser corrigidas ou removidas;

- **Regras incorretas:** regras que geram classificações enganosas afetam diretamente a eficácia do sistema, podendo levar a decisões erradas em contextos críticos, como classificações incorretas. A revisão e correção dessas regras por especialistas são essenciais para reduzir erros e aumentar a segurança do modelo.

A Figura 1.3 apresenta um exemplo de regras duplicadas, em que dois caminhos distintos resultam na mesma classificação final. Nesse caso, a Regra 17 ($D \leq 0.5 \wedge F \leq 0.5 \wedge G \leq 0.5 \wedge O \leq 0.5 \wedge C > 0.5 \wedge T > 0.5$) e a Regra 30 ($D \leq 0.5 \wedge F \leq 0.5 \wedge G \leq 0.5 \wedge O \leq 0.5 \wedge C > 0.5 \wedge T \leq 0.5$) complementam-se para atingir o mesmo resultado. Este cenário também evidencia que a remoção de uma regra pode levar à criação de novas duplicações, demandando ajustes adicionais. Embora tais redundâncias não comprometam diretamente a acurácia do modelo, elas impactam negativamente sua explicabilidade. Essa duplicação desnecessária sugere que o modelo poderia ser simplificado com regras mais claras e únicas, sem prejuízo à precisão das classificações. Além disso, essas redundâncias aumentam a complexidade do modelo, dificultando sua interpretação e reduzindo sua aplicabilidade prática em ambientes clínicos.

Além disso, a qualidade e representatividade dos dados utilizados para treinar esses modelos são fatores críticos. Dados enviesados ou mal representados afetam diretamente a capacidade

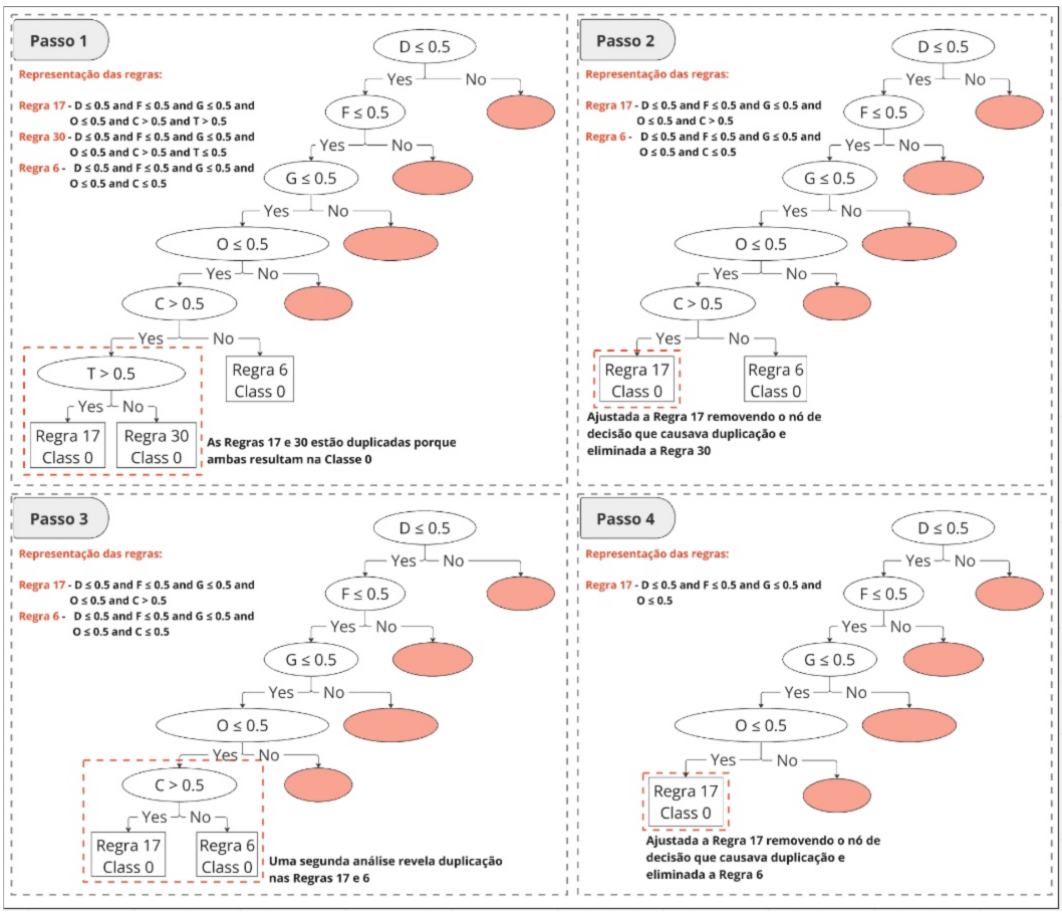

Figura 1.3: Exemplo de cenário de regras duplicadas.

de generalização, resultando em modelos que não atendem às demandas reais de ambientes clínicos. Assim, torna-se indispensável o desenvolvimento de métodos rigorosos para revisar, simplificar e validar as regras de decisão geradas. Tais regras podem ser identificadas analisando o número de *tokens* consumidos por cada transição representada por uma regra. Por exemplo, uma regra que classifica apenas uma única instância pode indicar uma regra específica.

Embora modelos baseados em DT e RF ofereçam maior explicabilidade do que métodos mais complexos, sua eficácia depende de abordar os problemas de regras redundantes, específicas e incorretas. Ignorar essas questões compromete a confiança no sistema e limita sua aplicação prática em áreas sensíveis como a saúde.

Para superar essas limitações, métodos formais emergem como uma abordagem promissora para análise e validação. CPN oferece ferramentas para lidar com problemas como redundâncias e inconsistências em regras de decisão, proporcionando maior transparência e precisão ao processo decisório, devido à sua capacidade de modelar, simular e verificar sistemas complexos.

Por meio da simulação de modelos CPN, é possível identificar e eliminar regras duplicadas, reduzir *overfitting* e corrigir classificações enganosas. Além disso, a abordagem baseada em CPN permite validar o sistema de forma iterativa, garantindo maior aderência aos rigorosos padrões de confiabilidade exigidos em sistemas críticos. Assim, o uso de CPN não apenas simplifica a estrutura dos modelos, mas também melhora sua explicabilidade, aumentando a confiança dos profissionais em sua aplicação prática.

1.3 Objetivos

O objetivo principal desta tese é propor e implementar um método baseado em CPN para melhorar a explicabilidade e identicar regras problemáticas de sistemas críticos baseados em AM na área da saúde, com foco específico em modelos de DT e RF. Os objetivos específicos estabelecidos neste trabalho são descritos a seguir:

1. desenvolver o método RULEXTRACT/CPN, que automatiza a extração de regras, a geração de modelos CPN, bem como a análise e o ajuste das regras de decisão produzidas por modelos DT e RF;

2. identificar e corrigir problemas em regras de decisão, como regras redundantes, regras específicas e regras incorretas;

3. desenvolver um sistema com base no método proposto, que reduza a necessidade de conhecimento especializado em CPN; e

4. validar o método proposto em estudos de caso na área da saúde, com foco em diagnósticos de COVID-19 e Influenza.

O método desenvolvido nesta tese utiliza CPN para modelar o conhecimento presente em

modelos de DT e RF, permitindo uma análise automática e detalhada das regras de decisão por meio de simulações. Os relatórios gerados viabilizam a identificação de regras redundantes e inconsistências, facilitando a simplificação e o ajuste dos modelos.

Além disso, para promover a adoção do método, foi criado um sistema que integra uma interface gráfica amigável. Essa solução permite que usuários sem conhecimento prévio em métodos formais analisem e ajustem regras de decisão de maneira prática e eficiente, ampliando o alcance e a utilidade do método em ambientes críticos.

Por fim, estudos de caso realizados no contexto da saúde validam o impacto do método em diagnósticos de COVID-19 e Influenza, destacando como ajustes nas regras de decisão influenciam positivamente a explicabilidade, a acurácia e a confiança nos sistemas de suporte à decisão.

1.4 Contribuições

Nesta tese, são apresentadas contribuições para as áreas de AM e modelagem formal, fundamentadas por uma base metodológica construída a partir de um estudo de mapeamento sistemático. Esse estudo foi conduzido seguindo as diretrizes de Petersen *et al.* [90], para identificar lacunas na literatura e orientar o desenvolvimento do trabalho. Observou-se que grande parte das pesquisas em AM concentra-se na validação de modelos, como redes neurais, DT e RF, utilizando métodos formais. Contudo, há uma ausência de estudos que abordem sistemas críticos ou discutam metodologias específicas para esse contexto. Assim, uma das contribuições desta tese é preencher essa lacuna ao explorar a integração de métodos formais e AM aplicados a sistemas críticos. As principais contribuições desta base metodológica incluem:

- a discussão sobre os métodos formais para apoiar o desenvolvimento de sistemas críticos baseados em AM;
- resultados do uso de métodos formais e AM, destacando desafios e limitações;
- identificação de oportunidades de pesquisa emergentes, enriquecendo a área de estudo multidisciplinar.

Além disso, foi desenvolvido um método baseado em simulação para analisar as regras de decisão em modelos de DT e RF. Para a implementação desse método, foram utilizadas tecnologias web e o arcabouço Access/CPN. O sistema resultante, denominado RuleXtract/CPN, permite a geração e o ajuste automático de modelos CPN, com o objetivo de melhorar tanto a explicabilidade quanto a acurácia dos modelos de AM. A simulação de modelos CPN facilita a identificação de regras duplicadas e, com isso, é possível eliminá-las, reduzindo a redundância e simplificando o processo de tomada de decisão. As principais contribuições deste trabalho incluem:

- discutir o uso de CPN para apoiar o desenvolvimento de sistemas críticos e aumentar a explicabilidade e a precisão dos modelos de AM;
- apresentar um método baseado em CPN para analisar regras de decisão duplicadas e problemáticas, com foco nos modelos de DT e RF;
- incluir o *RIPPER - Repeated Incremental Pruning to Produce Error Reduction* (JRip) na implementação para possibilitar uma comparação entre modelos baseados em regras e aqueles fundamentados em DT, avaliando suas diferenças em termos de explicabilidade, acurácia e redundância de regras;
- desenvolver um sistema para implementação do método; e
- experimentar o método em aplicações de saúde voltadas para o rastreamento de COVID-19 e Influenza, apontando desafios e oportunidades;

1.5 Metodologia

A metodologia definida para esta tese é ilustrada na Figura 1.4 e foi conduzida por meio de um processo estruturado em várias etapas, cada uma com um foco específico para alcançar os objetivos propostos. O processo começou com uma análise preliminar da literatura, na qual foram examinadas as principais teorias relacionadas ao AM e identificadas as limitações existentes.

Na etapa seguinte, foi realizado um mapeamento sistemático da literatura, com o objetivo

de investigar a aplicação de métodos formais para aprimorar sistemas críticos que utilizam AM. Para tanto, foram estabelecidos critérios de inclusão e exclusão, selecionadas bases de dados acadêmicas e realizadas buscas estruturadas para identificar estudos pertinentes. Os resultados desta busca foram cuidadosamente analisados e sintetizados, proporcionando uma visão consolidada sobre o tema.

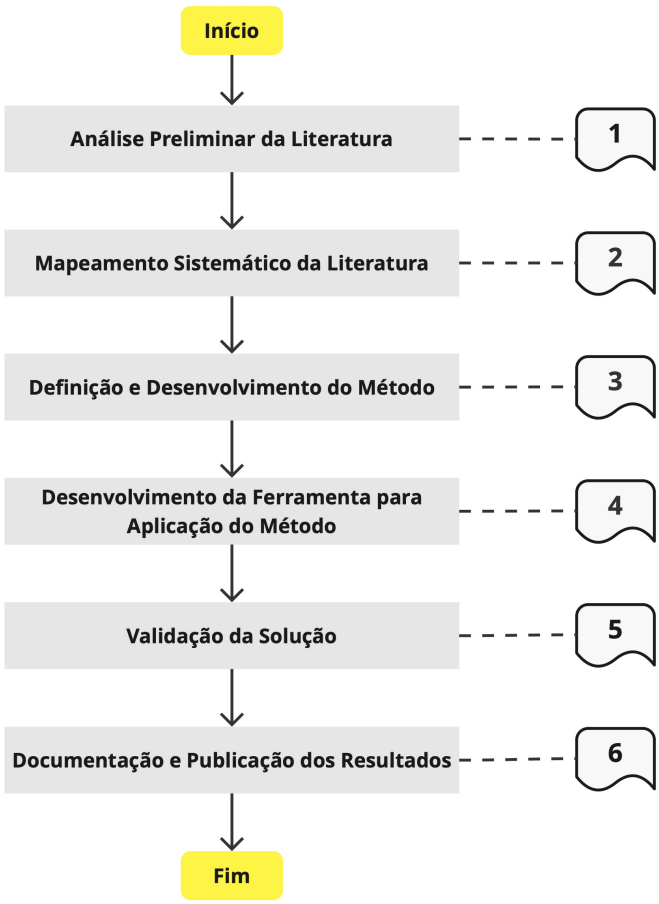

Figura 1.4: Etapas para a metodologia para modelagem e análise formal.

Com base nos conhecimentos adquiridos nas fases anteriores, a terceira etapa concentrou-se na definição e desenvolvimento do método. Nessa fase, foram delineados os processos e as técnicas que compõem o método, com o objetivo de aprimorar a explicabilidade e a confiabilidade de sistemas críticos baseados em AM. O método segue uma sequência de etapas: implementação do modelo de AM, extração automática de regras de decisão, transformação automática para o modelo CPN, análise das regras de decisão, ajuste do modelo CPN e,

finalmente, análise da acurácia do modelo.

Na quarta etapa, foi desenvolvida uma ferramenta para facilitar a aplicação do método baseado em simulação, automatizando seu uso em sistemas críticos. O desenvolvimento da ferramenta incluiu a criação de uma interface gráfica de usuário e a incorporação de funcionalidades essenciais para a execução do método.

A quinta etapa envolveu a validação da solução proposta, na qual a ferramenta desenvolvida foi aplicada a cenários práticos, utilizando dados de sistemas críticos na área da saúde, especificamente para diagnósticos de Covid-19 e Influenza. A validação foi conduzida para avaliar a eficácia em termos de melhoria da explicabilidade, confiança no funcionamento e acurácia dos sistemas de AM.

Por fim, na sexta etapa, todos os achados, metodologias, análises e conclusões foram documentados e preparados para publicação, garantindo que os resultados da pesquisa fossem amplamente compartilhados com a comunidade científica e prática, contribuindo para o avanço do conhecimento na área.

1.6 Organização do Documento

Este documento está estruturado em seis capítulos descritos a seguir:

- no Capítulo 2, é apresentado o embasamento teórico, abordando conceitos de AM, incluindo algoritmos de DT e RF. Além disso, são discutidos conceitos relacionados a métodos formais e CPN;
- no Capítulo 3, é descrito o estudo de mapeamento sistemático, explorando a interseção entre métodos formais e AM, e identificando tendências e oportunidades de pesquisa;
- no Capítulo 4, é apresentado um método baseado em simulação que utiliza CPN para analisar as regras de decisão dos modelos DT e RF;
- no Capítulo 5, são detalhados os resultados dos experimentos realizados com bases de dados relacionadas à COVID-19 e Influenza;

- no Capítulo 6, são discutidas as contribuições, limitações do método e possíveis ameaças à validade dos resultados obtidos;
- no Capítulo 7, são apresentadas as conclusões e direções futuras de pesquisa.

Capítulo 2

Embasamento Teórico

Neste capítulo, são apresentados os conceitos fundamentais das principais técnicas utilizadas no desenvolvimento do método baseado em simulação e nos experimentos descritos nos Capítulos 5 e 6. Inicialmente, abordam-se os conceitos essenciais sobre modelos de Aprendizado de Máquina (AM) empregados, especificamente as árvores de decisão (*Decision Tree - DT*) e florestas aleatórias (*Random Forest - RF*), além dos métodos formais, com destaque para as redes de Petri coloridas (Coloured Petri Nets - CPN). O objetivo é fornecer uma visão geral e introdutória sobre AM e métodos formais, estabelecendo as bases necessárias para a compreensão do estudo de caso e da metodologia de pesquisa.

2.1 Aprendizado de Máquina

Sistemas críticos podem se beneficiar do uso de AM para identificar padrões e tomar decisões com base nos dados coletados, eliminando a necessidade de programação explícita para tarefas específicas [136]. Desenvolvedores treinam algoritmos de AM para analisar e reconhecer padrões nos dados de treinamento, permitindo que esses algoritmos façam previsões ou decisões informadas sobre novos dados. Consequentemente, o AM tem aplicações significativas em áreas como saúde.

Diferentes algoritmos de AM exibem características e parâmetros únicos, permitindo sua classificação com base na linguagem de descrição, modo, paradigma e método de aprendi-

zado [136]. Os algoritmos geralmente dependem de métodos de aprendizado como aprendizado supervisionado, não supervisionado e por reforço [106]. Eles diferem nos dados de treinamento e no processo de aprendizado subsequente.

O aprendizado supervisionado, uma das abordagens mais importantes de AM, consiste em treinar um modelo utilizando um conjunto de dados previamente rotulado, em que as respostas corretas já estão definidas [33]. O objetivo do modelo é identificar padrões nos dados e realizar previsões precisas ao ser aplicado a informações não vistas anteriormente, podendo ser aplicado a tarefas como classificação e regressão. Por exemplo, o aprendizado supervisionado envolve a construção de um modelo a partir de instâncias de treinamento rotuladas, com o objetivo de aprender uma função que mapeie entradas para saídas. Este método visa desenvolver um modelo preditivo que possa rotular dados não vistos, otimizando sua capacidade de generalização e minimizando o erro nas previsões.

Por outro lado, o aprendizado não supervisionado considera dados não rotulados para identificar padrões e estruturas intrínsecas nos conjuntos de dados [120]. Esta categoria de algoritmos é comumente usada para tarefas como *clustering*, permitindo a descoberta de relações e organizações implícitas nos dados. Dessa forma, contribui para uma compreensão mais profunda das características fundamentais do conjunto de dados.

Considerando outra abordagem, o aprendizado por reforço consiste em um agente aprendendo a tomar decisões sequenciais interagindo com um ambiente dinâmico [82]. O agente executa ações e recebe *feedback* por meio de recompensas ou penalidades para aprender a realizar ações que maximizem as recompensas ao longo do tempo. Esta abordagem é frequentemente aplicada em problemas de tomada de decisão sequencial, como jogos e robótica.

Os desenvolvedores treinam modelos de AM usando algoritmos que analisam dados e identificam padrões. Consequentemente, o AM é relevante em várias áreas, incluindo robótica e saúde [74]. Essa tese foca em modelos de AM implementados para realizar tarefas de classificação, utilizando especificamente os algoritmos de DT e RF. Esses algoritmos são detalhados na próxima subseção.

2.1.1 Árvore de Decisão

A DT é um algoritmo de aprendizado supervisionado adotado para muitas tarefas de classificação, representando decisões em uma estrutura de árvore. Existem vários algoritmos de DT, como ID3, C4.5, CART, FT, BFTree e LMT. Por exemplo, o algoritmo C4.5 é o sucessor do ID3, usando a razão de ganho de informação como critério de divisão [104]. Esse critério emprega o conceito de entropia para estimar a dificuldade de prever o atributo alvo. O algoritmo calcula a entropia de A (com domínio $a_1, a_2, \ldots, a_n$) da seguinte forma [8]:

$$A = -\sum_{i=1}^{n} p_i log_2(p_i). \tag{2.1}$$

Na Equação 1, p_i é a probabilidade de observar cada valor $a1, a2, \ldots, an$. Os passos fundamentais para a construção de uma DT são explicitados no Algoritmo 1 [33]. Como entrada, o algoritmo recebe o conjunto de dados D, enquanto a saída é uma DT. Verifica-se então se a condição de parada foi atingida. Se necessário, o conjunto de dados é dividido em partes menores e o atributo que maximiza alguma métrica de impureza é selecionado. No sétimo passo, a função GeraÁrvore é chamada recursivamente para cada subdivisão do conjunto de dados D.

Algorithm 1 Algoritmo para construção de uma DT

Entrada: Um conjunto de treinamento D=(x_i,y_i), i=1,...,n
Saída: Árvore de Decisão
/* **Função GeraÁrvore(D)** */
if critério de parada(**D**) = Verdadeiro **then**
    Retorna: um nó folha rotulado com a constante que minimiza a função perda;
end if
Escolha o atributo que maximiza o critério de divisão em **D**;
for partição dos exemplos D_i baseado nos valores do atributo escolhido **do**
    Induz uma subárvore Árvorei = GeraÁrvore(D_i);
end for
Retorna: Árvore contendo um nó de decisão baseado no atributo escolhido, e descendentes Árvorei;

A DT baseia-se na estratégia de divisão e conquista, consistindo de nós de decisão e nós folha. Um nó de decisão especifica um teste em um dos atributos, enquanto um nó folha representa o valor da classe [101]. O algoritmo classifica as instâncias do nó raiz até o nó folha. Os modelos de DT oferecem várias vantagens, como facilidade de interpretação, preparação mínima dos dados, capacidade de lidar com problemas multiclasse e a capacidade de gerenciar dados numéricos e categóricos.

2.1.2 Floresta Aleatória (*Random Forest -RF*)

Além dos algoritmos de DT, a RF é uma técnica que combina várias DT para melhorar a precisão dos modelos. Uma RF é construída criando várias DTs em diferentes subconjuntos do conjunto de dados e agregando os resultados para melhorar a precisão preditiva e mitigar o *overfitting*. O algoritmo usado para criar cada árvore na RF é o mesmo utilizado em DT, com a diferença de que a construção é adaptada para introduzir maior diversidade entre as árvores. As principais diferenças estão na utilização de [32]:

- Construção com amostras aleatórias: Cada árvore é treinada utilizando uma seleção aleatória de dados do conjunto original, feita com reposição. Como resultado, algumas amostras podem aparecer mais de uma vez no mesmo conjunto de treinamento, enquanto outras não são incluídas, compondo o grupo denominado como "*Out of the Bag*";

- Uso de subconjuntos de atributos: Em cada nó, o modelo considera apenas um subconjunto específico e aleatório de variáveis para determinar a melhor divisão. Essa abordagem reduz a semelhança entre as árvores, aumentando a diversidade e a robustez do modelo;

- Crescimento total das árvores: Ao contrário das DTs convencionais, as árvores na RF são expandidas até atingirem o tamanho máximo possível, sem aplicar técnicas de poda, de forma a explorar completamente os padrões contidos nos dados.

Essas diferenças introduzem diversidade no conjunto de árvores e aumentam a capacidade de generalização do modelo. As previsões individuais das árvores são, então, combinadas para produzir uma única previsão final. Esse processo pode ser representado matematicamente

como [97]:

$$\hat{f}(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x) \tag{2.2}$$

Na Equação 2, $\hat{f}(x)$ é a previsão agregada da RF, N é o número de árvores na floresta, e $f_i(x)$ é a previsão da i-ésima árvore para a entrada x. Para problemas de classificação, a classe que recebe o maior número de votos entre as árvores é escolhida como a previsão final [63]. Para problemas de regressão, a média aritmética das previsões individuais é utilizada. Esse processo de votação majoritária contribui para aumentar a precisão do modelo e minimizar o risco de *overfitting*, uma vez que os erros de uma árvore podem ser compensados por outras. Além disso, essa abordagem garante que o RF seja um modelo eficiente, especialmente em contextos que envolvem conjuntos de dados extensos e complexos [97].

A Figura 2.1 ilustra o processo de construção e funcionamento de uma RF. O conjunto de dados original é dividido em amostras aleatórias (com reposição) para treinar diferentes DT. As amostras que não são selecionadas podem ser usadas para validação interna. Finalmente, os resultados das árvores individuais são combinados para gerar a previsão final (Y).

A RF oferece vantagens significativas em relação às DTs individuais, incluindo maior precisão, resistência a *outliers*, capacidade de lidar com grandes conjuntos de dados e um risco reduzido de *overfitting* [32]. Por essas razões, a RF é amplamente utilizada em várias tarefas de AM, especialmente em contextos onde há muitas variáveis ou dados complexos [10].

2.1.3 Comparação entre Algoritmos de Aprendizado de Máquina

Nesta subseção, é realizada uma análise comparativa entre alguns dos algoritmos de AM utilizados na literatura, incluindo DT, RF, JRIP e redes neurais. A comparação é baseada nas principais vantagens e desvantagens desses algoritmos, com ênfase em sua aplicabilidade em sistemas críticos. A análise busca evidenciar como as características intrínsecas de cada algoritmo podem influenciar sua adequação para contextos específicos. A Tabela 2.1 ilustra as características fundamentais de cada algoritmo, destacando como suas vantagens e desvantagens impactam sua aplicabilidade em sistemas críticos.

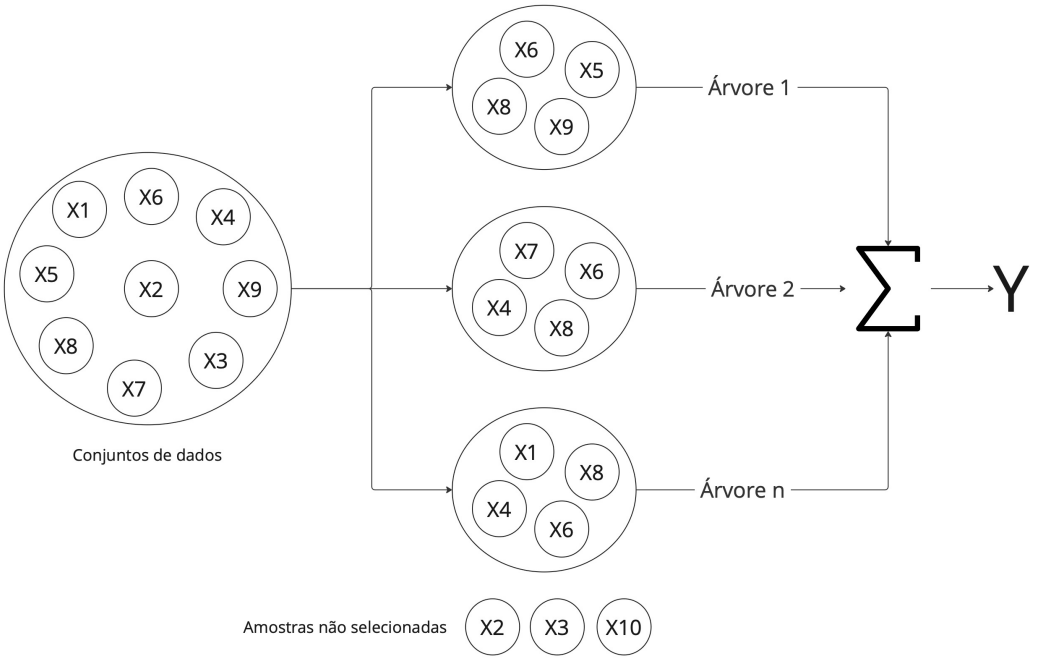

Figura 2.1: Algoritmo Random Forest [32]

As DT são amplamente reconhecidas por sua simplicidade e explicabilidade, características que tornam o processo de tomada de decisão mais intuitivo e fácil de interpretar. Essa transparência as torna especialmente adequadas para sistemas críticos, onde a compreensibilidade é essencial [93]. Contudo, sua sensibilidade a variações nos dados e a propensão ao *overfitting* em árvores profundas limitam sua robustez em cenários mais complexos [72].

Para superar essas limitações, as RF foram desenvolvidas, combinando os resultados de várias árvores para reduzir o risco de *overfitting*. Essa abordagem se mostra altamente eficaz em tarefas de classificação que envolvem grandes volumes de dados [32]. Apesar da elevada precisão, a explicabilidade reduzida e o alto custo computacional podem limitar sua aplicação em sistemas críticos que demandam decisões transparentes [88].

Em contraste, o algoritmo JRIP, baseado na indução de regras, apresenta uma solução mais simples, sendo ideal para conjuntos de dados pequenos ou moderados [25]. No entanto, sua incapacidade de capturar relações complexas pode comprometer o desempenho em cenários que exigem maior sofisticação analítica.

Tabela 2.1: Comparação de Algoritmos de Aprendizado de Máquina

Algoritmo	**Vantagens**	**Desvantagens**
DT	• Fácil de interpretar e visualizar. • Requer pouca preparação dos dados.	• Propensas ao *overfitting* em árvores profundas. • Sensíveis a variações nos dados.
RF	• Reduz o *overfitting* ao combinar múltiplas árvores; • Resistente a *outliers* e dados ruidosos. • Alta precisão em tarefas de classificação.	• Menos interpretável devido à complexidade do modelo. • Alto custo computacional para construção de várias árvores.
JRIP	• Gera regras simples e compreensíveis. • Bom desempenho em conjuntos de dados pequenos ou moderados.	• Menor precisão em problemas complexos ou de alta dimensionalidade. • Pode não capturar relações complexas nos dados.
RN	• Altamente eficazes em modelar padrões complexos e não lineares. • Excelente desempenho em grandes volumes de dados. • Flexíveis e aplicáveis em diferentes domínios	• Difíceis de interpretar (caixa-preta). • Requerem grandes recursos computacionais para treinamento. • Tendem a superestimar padrões em dados pequenos.

As DT, RF, JRIP e redes neurais apresentam características distintas que as tornam adequadas para diferentes cenários, com vantagens e limitações específicas. No contexto de sistemas críticos, como a saúde, onde a transparência e a explicabilidade são indispensáveis, as DT se destacam como uma escolha natural devido à sua simplicidade e clareza. No entanto, sua limitação em lidar com conjuntos de dados mais complexos e sua suscetibilidade ao *overfitting* motivam o uso de abordagens complementares.

As RF oferecem uma solução robusta para superar essas limitações, combinando a precisão de múltiplas árvores e sendo altamente eficazes em tarefas que envolvem grandes volumes de dados. Ainda assim, sua explicabilidade reduzida pode ser uma barreira em sistemas que demandam explicações claras para decisões tomadas. Nesse cenário, a integração de métodos formais, como CPN, surge como uma estratégia promissora. Ao transformar as regras de decisão de DT e RF em representações formais, o método permite analisar e ajustar os modelos, eliminando redundâncias e melhorando a explicabilidade, sem comprometer a precisão.

2.2 Métodos Formais

Métodos formais são técnicas baseadas em matemática para especificação, desenvolvimento e verificação de sistemas de *software* e *hardware*. Eles fornecem uma estrutura rigorosa para descrever o comportamento de sistemas, permitindo um raciocínio preciso e sem ambiguidades sobre a correção, segurança e desempenho de um sistema. Elas são especialmente importantes em sistemas críticos, nos quais falhas podem resultar em consequências graves, como perda de vidas, danos ambientais ou grandes prejuízos econômicos [131]. Métodos formais permitem a análise e verificação de propriedades essenciais dos sistemas, como a segurança [5, 112].

Esses métodos utilizam linguagens formais para descrever as características e comportamentos desejados dos sistemas. Por exemplo, os modeladores frequentemente utilizam lógica temporal para articular propriedades dependentes do tempo, essenciais para sistemas que respondem a estímulos externos em momentos específicos [7]. Além disso, métodos formais possibilitam a modelagem de sistemas complexos e não determinísticos, proporcionando

uma abordagem sistemática para verificar se um sistema atende aos requisitos especificados.

Os métodos formais podem ser aplicados em várias etapas do ciclo de desenvolvimento de *software*, incluindo a definição de requisitos e a validação de propriedades específicas do sistema. Entre as técnicas disponíveis estão a verificação automática de modelos (*model checking*), a demonstração por teoremas (*theorem proving*) e a análise estática do código. Essas abordagens auxiliam na identificação precoce de falhas no projeto, reduzindo custos e aprimorando a qualidade do *software* final. Por exemplo, a lógica de árvores computacionais (*computation tree logic*) [111], CPN [56] e teorias de satisfatibilidade modular (*Satisfiability Modulo Theories - SMT*) [132] são métodos formais amplamente utilizados para aumentar a confiança em sistemas complexos.

CPN foi escolhido para definir o método baseado em simulação devido à sua eficácia na análise de sistemas concorrentes e complexos. CPN integra a teoria de redes de Petri com uma linguagem de programação funcional baseada na *Standard Modeling Language (SML)* [52]. CPN oferece uma extensão da linguagem de programação SML, permitindo a implementação de conceitos importantes, como elementos de ligação.

2.2.1 Redes de Petri

O formalismo das redes de Petri foi criado pelo matemático alemão Carl Adam Petri em 1962. As redes de Petri são uma importante ferramenta de modelagem de sistemas, pois permitem avaliar a estrutura e o comportamento do sistema, resultando em melhorias ou mudanças. Elas podem ser usadas no contexto de sistemas concorrentes, assíncronos e distribuídos. Sistemas concorrentes são caracterizados pela execução simultânea de múltiplos processos que interagem entre si. Já os sistemas assíncronos possuem processos paralelos e não determinísticos, enquanto os sistemas distribuídos envolvem a interação de diversos componentes espalhados em diferentes locais [80].

Um modelo de redes de Petri é um grafo direcionado bipartido composto por lugares, transições, arcos direcionados e marcações [83]. Os lugares são usados para representar, por exemplo, estados parciais e são representados por círculos. As transições representam, por exemplo, eventos e têm o formato de barras. Os arcos são representados por setas que conec-

tam lugares e transições (nunca lugares a lugares ou transições a transições). As marcações indicam a quantidade de fichas (*tokens*) associadas aos lugares. Quando uma transição é disparada, as fichas são removidas dos lugares de entrada e novas fichas são geradas para os lugares de saída.

A dinâmica das redes de Petri permite a modelagem precisa de sistemas complexos, especialmente na representação de comportamentos concorrentes e paralelos. As fichas que se movem por meio dos lugares e transições fornecem uma visualização clara do fluxo de atividades e dos estados do sistema. Essa capacidade de representar e analisar o fluxo de controle e de dados de forma simultânea torna as redes de Petri uma ferramenta relevante para a análise de desempenho, detecção de *deadlocks* e verificação de propriedades de sistemas distribuídos e assíncronos.

2.2.2 Redes de Petri Coloridas

CPN é uma extensão das redes de Petri tradicionais, considerada uma rede de Petri de alto nível, desenvolvida no início da década de 1980 por Kurt Jensen em sua tese de doutorado [51]. A diferença entre redes de Petri tradicionais e CPN está no uso de componentes adicionais para modelar sistemas, como hierarquia. Enquanto as redes de Petri tradicionais usem apenas transições, lugares e arcos, as CPN usam cores, que são tipos de dados. Esta abordagem permite que os modelos de sistemas complexos sejam representados de forma mais clara e simples. A cor, ou tipo de dado, pode ser usada para representar qualquer atributo de uma entidade, como tipo, prioridade e estado.

Utilizou-se CPN para definir o método devido à sua eficácia na análise de sistemas concorrentes e complexos. CPN combina a teoria de redes de Petri com os recursos de uma linguagem de programação funcional, especificamente CPN ML [52]. A linguagem CPN ML é uma extensão do *Standard Modeling Language* (SML), desenvolvida para tornar a modelagem de redes de Petri mais intuitiva, permitindo que os usuários criem modelos de redes de Petri mais sofisticados e complexos. Além disso, a linguagem CPN ML também oferece aos usuários um conjunto de ferramentas para ajudar a construir e gerenciar modelos de redes de Petri. Estas ferramentas incluem editores de modelos, ferramentas de análise, ferramentas de simulação, ferramentas de verificação, ferramentas de otimização e ferramentas

de validação.

Um *módulo de rede de Petri colorida* é um tupla CPN_M = (P, T, A, Σ, V, C, G, E, I, T_{sub}, P_{port}, PT) [51, 53]:

1. P é um conjunto finito de lugares.
2. T é um conjunto finito de transições, tal que $P \cap T$ = θ.
3. $A \subseteq P \times T \cup T \times P$ é um conjunto de arcos direcionados.
4. Σ é um conjunto finito e não vazio de cores.
5. V é um conjunto finito de variáveis tipadas, tal que $Type[v] \in \Sigma$ para todas as variáveis $v \in V$.
6. $C : P \rightarrow \Sigma$ é uma função de conjunto de cores que atribui um conjunto de cores a cada lugar.
7. $G : T \rightarrow EXPR_V$ é uma função de guarda que atribui uma guarda a cada transição t, tal que $Type[G(t)] = Bool$.
8. $E : A \rightarrow EXPR_V$ é uma função de expressão de arco que atribui uma expressão de arco a cada arco a, tal que $Type[E(a)] = C(p)_{MS}$, onde p é o lugar conectado ao arco a.
9. $I : P \rightarrow EXPR_\theta$ é uma função de inicialização que atribui uma expressão de inicialização a cada lugar p, tal que $Type[I(p)] = C(p)_{MS}$.
10. $T_{sub} \subseteq T$ é um conjunto de transições de substituição.
11. $P_{port} \subseteq P$ é um conjunto de lugares de porta.
12. $PT : P_{port} \rightarrow$ IN, OUT, I/O é uma função de tipo de porta que atribui tipos de porta a lugares.

Portanto, uma *rede de Petri colorida hierárquica* é uma tupla com quatro elementos CPN_H = (S, SM, PS, FS) [52]:

1. S é um conjunto finito de *módulos*. Cada módulo é um *módulo de Rede de Petri Colorida* s = ((P^s, T^s, A^s, Σ^s, V^s, C^s, G^s, E^s, I^s), T^s_{sub}, P^s_{port}, PT^s). É requerido que ($P^{s_i} \cup T^{s_i}$)$\cap$($P^{s_j} \cup T^{s_j}$) = θ para todos s_i, $s_j \in S$ tal que $i \neq j$.

2. $SM : T_{sub} \rightarrow S$ é uma função de *submódulo* que atribui um submódulo a cada transição de substituição, requerendo que a *hierarquia do módulo* seja acíclica.

3. PS é uma função de relação porta-soquete que atribui uma *relação porta-soquete* $PS(t) \subseteq P_{sock}(t) \times P^{SM(t)}_{port}$ a cada transição de substituição t, requerendo que $PT(p)$ = $PT(p')$, $C(p)$ = $C(p')$ e $I(p)\langle\rangle$ para todos (p, p') $\in PS(t)$ e todos $t \in T_{sub}$.

4. $FS \subseteq 2^P$ é uma família de *conjuntos de fusão* não vazios, tal que $C(p)$ = $C(p')$ e $I(p)\langle\rangle$ = $I(p')\langle\rangle$ para todos p, $p' \in fs$ e todos $fs \in FS$.

A ISO/IEC 15909, partes 1 e 2, apresenta conceitos, definições, notação gráfica e formato de transferência baseado em XML para redes de Petri de alto nível. A Figura 2.2 ilustra os elementos representados no modelo CPN de acordo com as definições e atributos especificados em um arquivo XML. Lugares, que representam, por exemplo, condições ou estados parciais, são elipses. Transições, que representam, por exemplo, eventos, são retângulos. Arcos, que conectam lugares e transições, são setas que mostram a direção do fluxo, seja de um lugar para uma transição ou vice-versa.

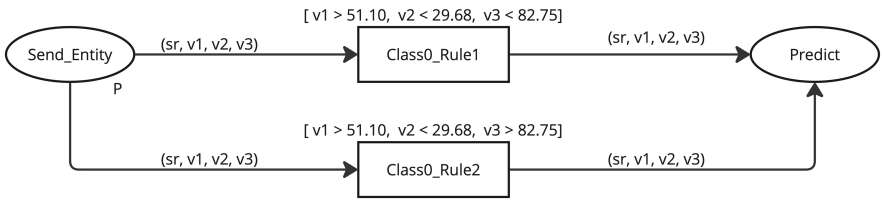

Figura 2.2: Metodologia proposta para modelagem e análise formal.

Em um modelo CPN, um lugar representa, por exemplo, uma condição ou estado parcial e pode conter *tokens* indicando a presença de uma condição específica. A *tag* `<place>` descreve um lugar no arquivo XML, incluindo elementos essenciais, como um identificador único, a marcação inicial e o tipo de *token*. A seguir está um exemplo de um lugar representado no arquivo XML:

```
<place id="P1" name="Place1">
  <initialMarking>
    <token>1</token>
  </initialMarking>
  <type>int</type>
  <posattr x="294.000000" y="114.0"/>
  <fillattr colour="White" pattern="" filled="false"/>
  <lineattr colour="Black" thick="1" type="Solid"/>
  <textattr colour="Black" bold="false"/>
</place>
```

Além desses itens essenciais, o XML pode conter *tags* que definem a posição do lugar no diagrama, a cor, o padrão de preenchimento, a espessura e o tipo da linha. Por exemplo, a *tag* `<posattr x="294.000000"y="114.0"/>` define a posição do lugar, enquanto a *tag* `<fillattr colour="White"pattern= filled="false"/>` especifica a cor e o padrão de preenchimento. A *tag* `<lineattr colour="Black"thick="1"type="Solid"/>` define a cor, a espessura e o tipo da linha do lugar, e a *tag* `<textattr colour="Black"bold="false"/>` especifica o estilo do texto.

Uma transição em um modelo CPN representa, por exemplo, um evento ou uma mudança de estado que pode ocorrer. A descrição de uma transição no arquivo XML é feita com a *tag* `<trans>`, que inclui elementos essenciais, como um identificador único, posição, cor, nome da transição (`<text>`) e condições de guarda (`<cond>`). Além desses itens essenciais, o XML pode conter *tags* que definem a posição da transição no diagrama (`<posattr>`), cor e padrão de preenchimento (`<fillattr>`), cor da linha, espessura e tipo (`<lineattr>`), e o estilo do texto associado à transição (`<textattr>`). As *tags* `<cond>`, `<time>`, `<code>` e `<priority>` também incluem elementos que definem posição, cor e detalhes adicionais:

```
<trans id="ID50505051" explicit="false">
```

```
        <posattr x="-182.0" y="84.0"/>
        <fillattr colour="White" pattern="" filled="false"/>
        <lineattr colour="Black" thick="1" type="solid"/>
        <textattr colour="Black" bold="false"/>
        <text>Class0_Rule1</text>
        <box w="100.000000" h="40.000000"/>
        <binding x="-182.0" y="84.0"/>
        <cond id="ID51515152">
          <text tool="CPN Tools" version="4.0.1">[v9 >
          0.5]</text>
        </cond>
        <time id="ID525252524">...</time>
        <code id="ID53535354">...</code>
        <priority id="ID54545455">...</priority>
</trans>
```

Os arcos conectam lugares e transições no modelo CPN, definindo a direção e as condições para o movimento dos tokens entre eles. A descrição de um arco no arquivo XML é feita com a *tag* `<arc>`. A seguir está um exemplo de como um arco é escrito:

```
<arc id="ID1413764052" orientation="PtoT" order="1">
    <posattr x="0.000000" y="0.000000"/>
    <fillattr colour="White" pattern="" filled="false"/>
    <lineattr colour="Teal" thick="1" type="Solid"/>
    <textattr colour="Teal" bold="false"/>
    <arrowattr headsize="1.200000" currentcyckle="2"/>
    <transend idref="ID1413640578"/>
    <placeend idref="ID1412415160"/>
    <annot id="ID1413764053">
      <posattr x="-306.500000" y="369.000000"/>
      <fillattr colour="White" pattern="Solid" filled="false"
```

```
12        />
13        <lineattr colour="Teal" thick="0" type="Solid"/>
14        <textattr colour="Teal" bold="false"/>
15        <text tool="CPN Tools" version="4.0.1">(sr,v1,v2,v3,v4)
16        </text>
17      </annot>
18  </arc>
```

A *tag* <arc id="ID14052"orientation="PtoT"order="1"> define um arco com um identificador único, especificando sua orientação (de lugar para transição ou vice-versa) e ordem. A posição do arco no diagrama é definida pela *tag* <posattr>, enquanto a cor e o padrão de preenchimento são determinados pela *tag* <fillattr>. A *tag* <lineattr> define a cor, espessura e tipo da linha do arco, e a *tag* <textattr> especifica o estilo do texto associado ao arco. O tamanho da cabeça da seta do arco é definido pela *tag* <arrowattr>. A transição conectada ao arco é referenciada pela *tag* <transend idref="ID1413640578"/>, e o lugar conectado ao arco é indicado pela *tag* <placeend idref="ID1412415160"/>. Anotações adicionais sobre o arco estão contidas na *tag* <annot>, enquanto o texto exibido no arco, que pode incluir variáveis ou informações adicionais, é definido pela *tag* <text>(sr,v1,v2,v3,v4)</text>.

Esta linguagem formal de modelagem é amplamente utilizada para analisar e garantir que sistemas complexos e não-determinísticos atendam às propriedades desejadas. Portanto, este formalismo é relevante para modelar e analisar modelos de AM. O *software CPN Tools* ou CPN IDE pode ser usado para desenvolver e analisar modelos CPN, fornecendo capacidades de edição, simulação e realização de análises de espaço de estados.

O uso do *software CPN Tools* ou CPN IDE para simulação oferece diversos benefícios. Ele permite a visualização dinâmica do comportamento do sistema, facilitando a identificação de potenciais problemas e a análise das propriedades do sistema em vários cenários. A simulação auxilia na validação do modelo por meio de uma observação detalhada, permitindo a detecção precoce de inconsistências ou erros. Além disso, facilita análises de desempenho, como a medição de tempos de resposta ou utilização de recursos em diferentes condições

operacionais. As simulações também permitem a experimentação com várias configurações e parâmetros sem alterar o sistema, oferecendo uma forma segura e econômica de testar hipóteses.

No entanto, essas ferramentas não são fáceis de usar e exigem uma curva de aprendizado relativamente acentuada para conduzir de maneira eficaz a modelagem e análise formal. Devido à complexidade das tarefas de modelagem e análise, existem ferramentas para gerenciar modelos CPN sem o uso de uma interface gráfica. O arcabouço Access/CPN permite que os usuários lidem com componentes do modelo, simulações e o código SML por meio de Java [129, 35]. Ele não se destina a substituir o CPN Tools como editor de modelos CPN, mas sim a permitir que pesquisadores e desenvolvedores conduzam experimentos com o formalismo CPN e o integrem no desenvolvimento de aplicações.

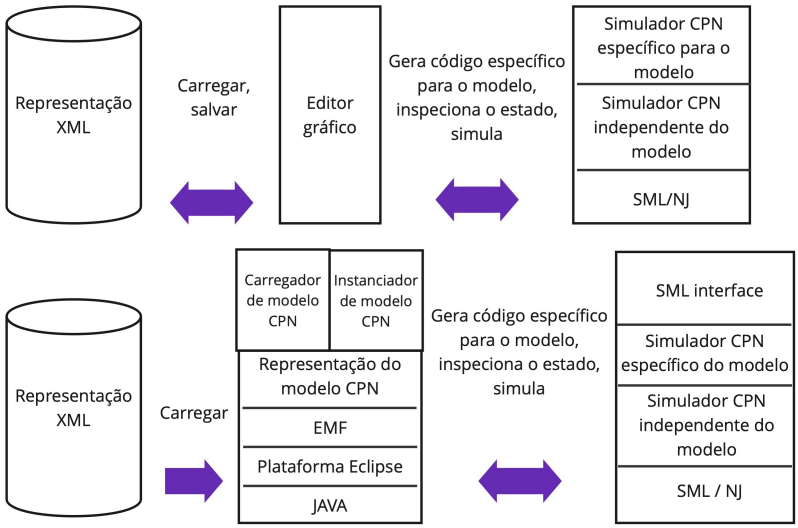

Figura 2.3: Arquitetura do CPN Tools (superior) e Access/CPN (inferior) *et al.*

Fonte: Westergaard e Kristensen (2009) [130] - Traduzido.

Na Figura 2.3, é ilustrada a arquitetura do CPN Tools (na parte superior) e como o Access/CPN complementa e substitui partes do CPN Tools (na parte inferior). Na parte superior da figura, o CPN Tools é composto por um editor gráfico que permite a construção interativa de um modelo CPN. Esse modelo é então enviado ao simulador para verificação de erros sintáticos e geração de código específico do modelo [130]. Esse código é utilizado para simular o modelo CPN, com os resultados sendo apresentados graficamente. O editor é capaz

de carregar e salvar modelos utilizando um formato XML.

Na parte inferior da figura, pode-se observar como o Access/CPN se integra, substituindo o editor gráfico por interfaces Java e SML. A interface Java oferece uma representação orientada a objetos dos modelos CPN, possibilitando a transmissão dessa representação para o simulador e a execução de simulações, além de permitir a inspeção do estado atual no simulador [129]. Ela inclui um carregador que pode importar modelos criados com o CPN Tools ou CPN IDE. A interface SML encapsula as estruturas de dados utilizadas no simulador, fornecendo uma interface para um modelo CPN que facilita a simulação rápida e eficiente, sendo útil para análises e outras aplicações que exigem pouca ou nenhuma interação do usuário [129, 35].

2.3 Considerações Finais

Neste capítulo, foram apresentados conceitos fundamentais sobre AM, focando especificamente em DT e RF. Explorou-se a construção e os benefícios das RFs, incluindo a melhoria na precisão e robustez das previsões através da combinação de múltiplas DTs. Além disso, foram discutidos os métodos formais, com ênfase nas CPN, e como essas ferramentas são essenciais para a modelagem e verificação de sistemas complexos.

Os detalhes sobre a estrutura e operação das CPNs, foram abordados para proporcionar uma compreensão clara de como essas redes podem ser utilizadas para modelar sistemas concorrentes e distribuídos. A integração da linguagem CPN ML e o uso do CPN Tools foram destacados, demonstrando como esses recursos auxiliam na análise e simulação de sistemas complexos.

Por fim, a arquitetura do Access/CPN foi explicada, mostrando como ele complementa o CPN Tools, permitindo a manipulação e simulação de modelos CPN usando interfaces Java e SML. A utilização dessas ferramentas e técnicas proporciona uma base sólida para a análise formal de modelos de AM, facilitando a detecção precoce de inconsistências e melhorando a qualidade dos sistemas desenvolvidos.

Esses conceitos e ferramentas estabelecem uma base essencial para a compreensão dos tópi-

cos abordados nos experimentos detalhados no Capítulo 5, proporcionando um embasamento necessário para o desenvolvimento e a aplicação do método proposto neste estudo.

Capítulo 3

Trabalhos Relacionados

Neste capítulo, são apresentados os trabalhos relacionados a esta pesquisa, analisados com base na metodologia de mapeamento da literatura. O mapeamento sistemático focou no uso de métodos formais para melhorar sistemas críticos baseados em Aprendizado de Máquina (AM). O objetivo foi fornecer uma visão abrangente do campo, identificando tendências emergentes, lacunas na pesquisa e novas oportunidades. Esse tipo de estudo é essencial para consolidar o conhecimento existente e direcionar futuras pesquisas para áreas que necessitam de maior desenvolvimento.

A combinação do rigor dos métodos formais com a adaptabilidade e as capacidades do AM oferece uma abordagem mais confiável e eficiente para o desenvolvimento de sistemas críticos. Essa convergência é crucial, especialmente em domínios onde a precisão e a segurança são fundamentais. Métodos formais fornecem garantias matemáticas sobre o comportamento dos sistemas, enquanto o AM permite a adaptação e a melhoria contínua com base em dados empíricos.

Para explorar o estado da arte nessa interseção, foi realizado um estudo de mapeamento sistemático da literatura, analisando mais de 240 artigos para investigar como os métodos formais têm sido utilizados para melhorar sistemas críticos baseados em AM. O foco foi identificar até que ponto soluções baseadas em métodos formais foram desenvolvidas e testadas para aprimorar a explicabilidade, a confiabilidade e o desempenho desses sistemas.

Nas seções seguintes, serão apresentados os resultados detalhados deste mapeamento, incluindo a análise das técnicas de AM mais utilizadas, os métodos formais empregados, as ferramentas desenvolvidas e os contextos críticos em que essas soluções têm sido aplicadas. A partir dessa análise, espera-se fornecer uma base sólida para pesquisadores e profissionais que buscam avançar no desenvolvimento de sistemas críticos mais seguros e confiáveis.

3.1 Metodologia de Pesquisa

Este mapeamento sistemático da literatura examina e sintetiza criticamente as pesquisas existentes sobre métodos formais aplicados em AM para sistemas críticos e cenários de aplicação específicos. A metodologia descreve as etapas realizadas no mapeamento sistemático da literatura e o desenvolvimento e estrutura do protocolo. A metodologia utilizada segue as diretrizes propostas por Petersen *et al.* [90]. Assim, o mapeamento sistemático da literatura foi estruturado da seguinte forma.

- Formulação das Perguntas de Pesquisa (PPs): definição de PPs para guiar o foco e os objetivos do estudo;
- Metodologia de Pesquisa: detalhamento dos métodos e abordagens específicos para conduzir a pesquisa, garantindo sua validade e confiabilidade;
- Processo de Seleção: estabelecimento de critérios de inclusão e exclusão para determinar estudos relevantes para o mapeamento sistemático da literatura, garantindo abrangência e relevância; e
- Extração e Síntese de Dados: coleta e consolidação de dados dos estudos selecionados para entender as tendências emergentes e padrões de pesquisa.

Cada etapa é descrita detalhadamente nas seções seguintes, delineando a metodologia utilizada neste mapeamento sistemático da literatura.

3.1.1 Perguntas de Pesquisa

As PPs foram definidas para assegurar uma revisão que esteja alinhada com o objetivo principal do mapeamento sistemático da literatura. Este mapeamento visa identificar, classificar e analisar estudos científicos focados na aplicação de métodos formais em AM no contexto de sistemas críticos. As PPs formuladas guiam a investigação, assegurando que os resultados sejam relevantes e contribuam para o campo de estudo. Foi definida a PP1 da seguinte forma:

- **PP 1.:** Qual é o estado da arte na interseção de métodos formais e AM?

A PP1 visa identificar aplicações específicas de AM, os métodos formais, ferramentas formais, e o contexto crítico em que essas técnicas se aplicam. Assim, foi definida três PPs secundárias:

- **PP1.1.:** Quais algoritmos específicos de AM foram usados em conjunto com métodos formais?
- **PP1.2.:** Quais métodos formais foram empregados em conjunto com AM?
- **PP1.3.:** Quais métodos e ferramentas formais foram identificados?
- **PP1.4.:** Em quais contextos críticos a combinação de métodos formais e AM foi aplicada?

Além disso, foi definida a PP2 da seguinte forma:

- **PP2.:** Qual é o nível de maturidade das soluções existentes na interseção de métodos formais e AM em contextos críticos?

A PP2 explora o avanço e o status de desenvolvimento das soluções que combinam métodos formais e AM no contexto de sistemas críticos, analisando se estão em estágios iniciais, em desenvolvimento ou já maduras e amplamente implementadas. Assim, definiu-se três PPs secundárias:

- **PP2.1.:** Qual é o impacto da evolução das publicações e citações em artigos científicos em AM e métodos formais em contextos críticos, e como esses fatores se correlacionam?

- **PP2.2.:** Quais contribuições de pesquisa os estudos sobre a convergência de métodos formais e AM forneceram?

Essas PPs fundamentaram o desenvolvimento do protocolo de revisão e estruturaram o processo do mapeamento sistemático da literatura. Elas guiaram a busca e seleção de estudos, assegurando rigor e objetividade, e contribuíram para a conclusão do estudo.

3.1.2 Método de Busca

O principal desafio na formulação da *string* de busca para a pesquisa em bases de dados foi abranger a diversidade desejada de estudos, mantendo um equilíbrio entre a amplitude, incluindo todos os estudos relevantes, e a precisão para garantir a eficiência do processo subsequente de seleção. Assim, adotou-se o protocolo *Population, Intervention, Comparison, Context, and Outcomes* (PICOC) utilizado na ferramenta Parsifal para incluir estudos pertinentes sem comprometer o escopo da busca. Este protocolo usa critérios PICOC para estruturar a *string* de busca. Esta abordagem definiu os aspectos cruciais do estudo, garantindo uma busca inclusiva o suficiente para cobrir estudos relevantes, apesar do potencial aumento no volume de trabalho para filtragem manual no processo de seleção. Após a implementação do protocolo estabelecido, definiu-se a *string* de busca, ilustrada na Figura 3.1.

Inicialmente, as bases de dados consultadas para a pesquisa incluíram a *ACM Digital Library*, *IEEE Xplore*, PubMed, *Web of Science* e Scopus, que foram escolhidas por sua coleção abrangente de artigos de periódicos e anais de conferências relacionados ao tópico de pesquisa. Após esta fase, realizou-se uma busca suplementar no Google *Scholar* para garantir a inclusão de quaisquer estudos relevantes que não estivessem presentes nas bases de dados principais ou encontrados devido às limitações das *strings* de busca usadas, maximizando assim a cobertura da revisão de literatura. Este tipo de busca suplementar é uma prática comum e bem-aceita para revisões de literatura [28] [113].

3.1.3 Processo de Seleção

Adotou-se uma metodologia de seleção de artigos em três etapas para a estratégia de busca em bases de dados neste estudo. A fase inicial consistiu em uma filtragem preliminar para

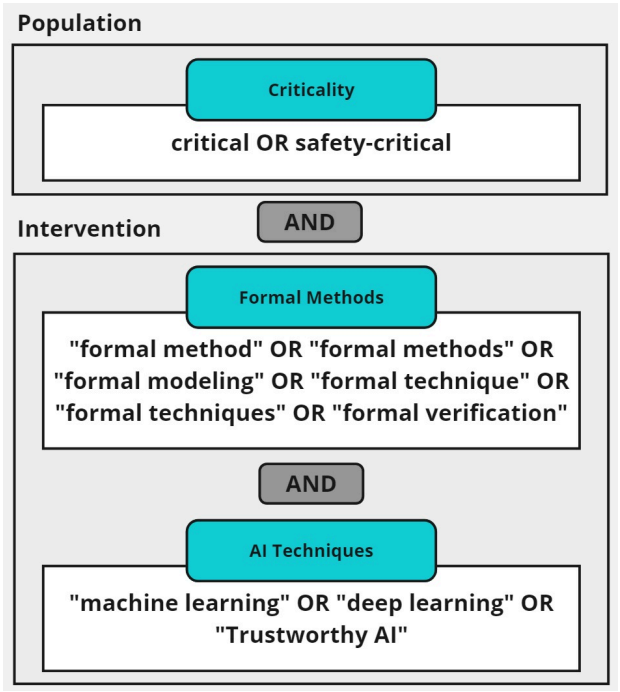

Figura 3.1: *String* de busca usada em bases de dados.

excluir artigos duplicados em diferentes bases de dados. Utilizou-se a plataforma Parsifal para facilitar o processo nesta etapa preliminar. Subsequentemente, aplicou-se critérios específicos de inclusão e exclusão para avaliar a adequação dos artigos aos objetivos do estudo. Os critérios de inclusão considerados foram os seguintes:

- artigos revisados por pares;
- disponibilidade do artigo completo;
- artigos que relatem o uso de AM e métodos formais em sistemas críticos;

Além disso, os critérios específicos de exclusão considerados foram os seguintes:

- artigos não revisados por pares, como teses;
- estudos indisponíveis em formatos acessíveis ou em idiomas diferentes do inglês;
- estudos publicados antes de 2011;

- artigos curtos com quatro páginas ou menos;
- estudos que não focam em sistemas críticos;
- estudos consistindo em uma revisão de literatura; e
- estudos que não apresentam estudos de caso ou discussões aprofundadas sobre sistemas críticos.

Realizou-se essa filtragem através de uma abordagem de leitura adaptativa, com pelo menos dois revisores avaliando cada artigo. A seleção teve como objetivo identificar estudos que abordassem a interseção entre métodos formais e AM em sistemas críticos, focando em pesquisas que fornecessem *insights* significativos para o avanço e a compreensão dessas tecnologias em contextos de alta criticidade. Este processo visou assegurar que apenas artigos relevantes e de alta qualidade fossem incluídos no mapeamento sistemático da literatura, enriquecendo assim a análise com contribuições fundamentais para o campo.

Resultados do Processo de Seleção: Na Figura 3.2 é apresentada uma visão geral do procedimento utilizado no mapeamento sistemático da literatura, detalhando as etapas metodológicas desde a concepção inicial até a análise final dos dados. Este diagrama destaca os principais componentes do processo, incluindo a busca e seleção de estudos, a aplicação de critérios de inclusão e exclusão, bem como a extração e síntese das informações. Fornece uma compreensão clara e organizada da abordagem metodológica adotada no estudo.

Busca por estudos primários

A busca por estudos primários cobriu várias bases de dados em 16 de julho de 2023. Como resultado dessa pesquisa, analisaram-se 240 artigos organizados em arquivos estruturados, como arquivos BibTeX, e posteriormente foram incluídos na ferramenta Parsifal.

Triagem preliminar

Aplicou-se o filtro da ferramenta Parsifal para remover artigos duplicados em diferentes bases de dados. Esse processo eliminou 55 artigos duplicados, deixando 185 artigos restantes.

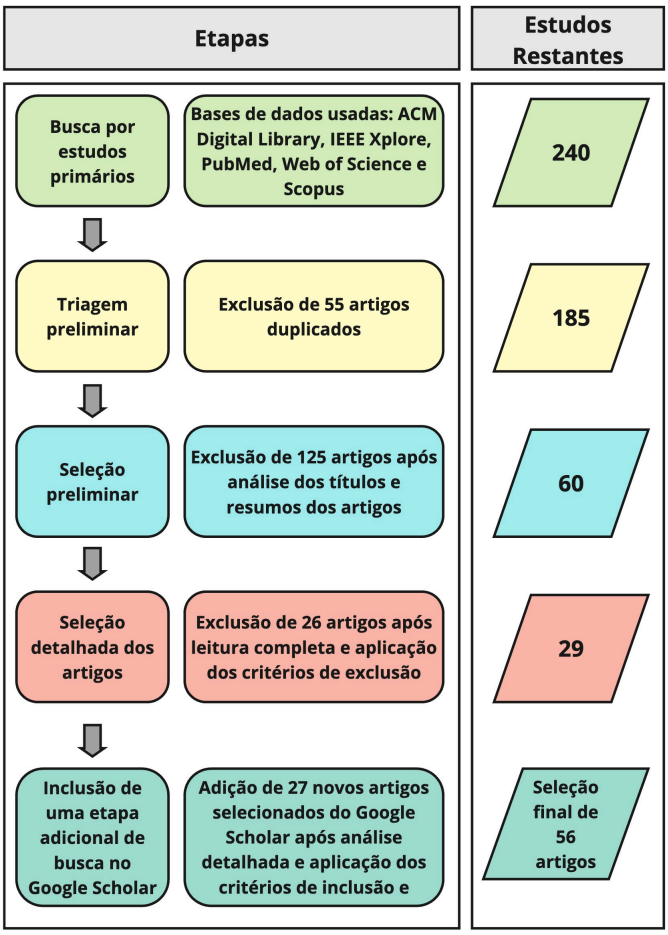

Figura 3.2: Visão geral do processo de busca e seleção do mapeamento sistemático da literatura.

Seleção preliminar

Após análise cuidadosa dos títulos e resumos dos artigos, considerando os critérios de exclusão estabelecidos, 60 artigos foram selecionados para prosseguir no estudo, enquanto 125 foram removidos.

Seleção detalhada dos artigos

Em seguida, completou-se a leitura de cada artigo, aplicando os critérios de exclusão estabelecidos. Após essa análise minuciosa, 29 artigos foram mantidos no mapeamento sistemático da literatura. Na Tabela 3.1 são apresentados os 29 artigos selecionados nesta etapa.

Tabela 3.1: Artigos aceitos com suas respectivas bases de dados.

Artigo	**Base de Dados**
Formal verification of input-output mappings of tree ensembles [119]	Scopus
DeepAuto: A first step towards formal verification of deep learning systems [69]	Scopus
Formal Verification of Neural Networks: a Case Study about Adaptive Cruise Control [27]	Scopus
Formal Verification of Tree Ensembles against Real-World Composite Geometric Perturbations [20]	Scopus
Certified reinforcement learning with logic guidance [43]	Scopus
Bounding Perception Neural Network Uncertainty for Safe Control of Autonomous Systems [128]	PubMed
Using Quantifier Elimination to Enhance the Safety Assurance of Deep Neural Networks [96]	WoS
An Inductive Synthesis Framework for Verifiable Reinforcement Learning [138]	WoS
Verifiably safe off-model reinforcement learning [36]	Scopus
Formal Verification of Random Forests in Safety-Critical Applications [117]	Scopus
A formal methods approach to interpretable reinforcement learning for robotic planning [66]	Scopus
Towards verifiable and safe model-free reinforcement learning [44]	Scopus
Safe reinforcement learning using probabilistic shields [49]	Scopus
An abstraction-based method to check multi-agent deep reinforcement-learning behaviors [78]	Scopus
Formal Verification of Neural Networks for Safety-Critical Tasks in Deep Reinforcement Learning [21]	Scopus
Safe Reinforcement Learning using Formal Verification for Tissue Retraction in Autonomous Robotic-Assisted Surgery [91]	Scopus
Verifiably safe exploration for end-to-end reinforcement learning [45]	Scopus
Counter-Example Guided Abstract Refinement for Verification of Neural Networks [26]	Scopus
Verification of Neural Networks for Safety and Security-critical Domains [40]	Scopus
Minimal-unsatisfiable-core-driven Local Explainability Analysis for Random Forest [71]	Scopus
Energy-Efficient Control Adaptation with Safety Guarantees for Learning-Enabled Cyber-Physical Systems [126]	Scopus
A Unified View of Piecewise Linear Neural Network Verification [11]	WoS
Abstract Layer for LeakyReLU for Neural Network Verification Based on Abstract Interpretation [75]	WoS
On Optimizing Back-Substitution Methods for Neural Network Verification [135]	IEEE
Selecting Stable Safe Configurations for Systems Modelled by Neural Networks with ReLU Activation [9]	IEEE
Flight Test of a Collision Avoidance Neural Network with Run-Time Assurance [19]	IEEE
PAC-Based Formal Verification for Out-of-Distribution Data Detection [92]	IEEE
An MILP Encoding for Efficient Verification of Quantized Deep Neural Networks [76]	IEEE
Empirical study on security verification and assessment of neural network accelerator [15]	IEEE

Inclusão de uma Etapa Adicional de Busca no Google *Scholar*

Finalmente, foi realizada uma busca no Google *Scholar*, e 27 novos artigos foram selecionados após uma análise detalhada e aplicação dos critérios de inclusão e exclusão. Esta etapa foi realizada para encontrar artigos que não estavam presentes nas bases de dados pesquisadas anteriormente ou que não apareceram nos resultados da *string* de busca utilizada. Após todas as análises, 56 artigos foram mantidos no mapeamento sistemático da literatura. Na Tabela 3.2 são apresentados os 27 artigos selecionados nesta etapa.

Tabela 3.2: Artigos indexados no Google Scholar e suas respectivas bases de dados.

Artigo	**Base de Dados**
NNLander-VeriF: A Neural Network Formal Verification Framework for Vision-Based Autonomous Aircraft Landing [103]	Springer
Formal verification of neural network controlled autonomous systems [115]	ACM
Formal Verification of Piece-Wise Linear Feed-Forward Neural Networks [29]	Springer
Verification of Neural Network Behaviour: Formal Guarantees for Power System Applications [61]	IEEE
The Marabou Framework for Verification and Analysis of Deep Neural Networks [59]	Springer
DeepCert: Verification of Contextually Relevant Robustness for Neural Network Image Classifiers [89]	Springer
Verification of image-based neural network controllers using generative models [60]	ARC
Formal Verification of Stochastic Systems with ReLU Neural Network Controllers [114]	IEEE
Framework for Formal Verification of AM Based Complex System-of-Systems [95]	Incose
Safety Verification of Neural Network Controlled Systems [18]	IEEE
An Abstraction-Based Framework for Neural Network Verification [31]	Springer
Robustness Verification of Swish Neural Networks Embedded in Autonomous Driving Systems [137]	IEEE
Formal Verification of Deep Neural Networks in Hardware [102]	IEEE
Neural Network Verification using Residual Reasoning [30]	Springer
Formal Verification for Safe Deep Reinforcement Learning in Trajectory Generation [22]	IEEE
Conformance verification for neural network models of glucose-insulin dynamics [64]	ACM
Star-Based Reachability Analysis of Deep Neural Networks [118]	Springer
VeRe: Verification Guided Synthesis for Repairing Deep Neural Networks [70]	ACM
PRIMA: General and Precise Neural Network Certification via Scalable Convex Hull Approximations [79]	ACM
Safety Verification and Robustness Analysis of Neural Networks via Quadratic Constraints and Semidefinite Programming [34]	IEEE
Evaluation of Neural Network Verification Methods for Air-to-Air Collision Avoidance [73]	ARC
Formal Verification and Development of Living Assistance System [1]	IEEE
Towards Verification of Neural Networks for Small Unmanned Aircraft Collision Avoidance [46]	IEEE
Case study: verifying the safety of an autonomous racing car with a neural network controller [47]	ACM
Verifying the Safety of Autonomous Systems with Neural Network Controllers [48]	ACM
Assured runtime monitoring and planning: Toward verification of neural networks for safe autonomous operations [134]	IEEE
Improving the Correctness of Medical Diagnostics Based on AM With Coloured Petri Nets [84]	IEEE

A análise da distribuição dos 56 artigos selecionados para este estudo revela uma predominância notável de certas bases de dados. A Scopus surge como a plataforma mais representativa, contribuindo com 18 artigos, o que corresponde a aproximadamente 32,14% do total. Logo atrás, a base de dados IEEE Xplore desempenha um papel crucial, fornecendo 17 artigos, ou 30,36% do conjunto. A Springer adiciona 7 artigos, representando cerca de 12,50%, enquanto a ACM fornece 6 artigos, totalizando 10,71%. A Web of Science, por sua vez, oferece 4 artigos, que representam 7,14% do total. As bases de dados ARC, com 2 artigos, contribuem com 3,57%, e tanto a PubMed quanto a Incose, com um artigo cada,

representam cerca de 1,79% cada, mostrando que, embora menos representativas, ainda forneceram contribuições valiosas para a amplitude e profundidade do mapeamento sistemático da literatura. Esta distribuição destaca a importância de acessar várias bases de dados para compreender de forma abrangente o domínio estudado.

3.1.4 Extração e Síntese de Dados

Após identificar todos os estudos iniciais, dados referentes à demografia foram extraídos e às PPs abordadas nos artigos com base nos detalhes fornecidos na Tabela 3.3. Cada pesquisador foi responsável por analisar o conteúdo completo dos estudos atribuídos, visando preencher adequadamente os campos de extração. Subsequentemente, as características (Pr) ilustradas na Tabela 3.3 foram detalhadas. Para os indicadores Pr1-Pr3, os dados coletados incluíram os títulos dos trabalhos e a origem das bases de dados das quais foram extraídos.

A PP1 adotou uma metodologia focada em três variáveis principais. No Pr4, o objetivo foi catalogar e compreender a aplicação de modelos de AM, incluindo técnicas como DT, análise de regressão, redes neurais e máquinas de vetor de suporte, focando em como esses modelos representam soluções e resolvem desafios complexos em ambientes críticos.

Em relação ao Pr5, a investigação focou no contexto crítico de aplicações que integram AM e métodos formais, analisando setores de alta criticidade, como controle de voo, veículos autônomos, sistemas de prevenção de colisões aéreas e aplicações médicas, que exigem altos padrões de confiabilidade e precisão.

Tabela 3.3: Resumo das propriedades da pesquisa.

ID	Propriedade	Formato/Valor	PP
Pr1-Pr3	ID da Publicação, título, fonte	Número, título do artigo, base de dados	-
Pr4	Modelos de AM	Por exemplo, Árvore de decisão, análise de regressão, redes neurais, SVM	PP1
Pr5	Contexto de aplicação crítica	Por exemplo, Controle de voo, carro, médico	PP1
Pr6	Tipos de MF suportados por AM	Por exemplo, Lógica Temporal Linear, verificação de modelos	PP1
Pr7-Pr8	Ano de publicação, número de citações	Número	PP2
Pr9	Tipo de pesquisa empírica	Por exemplo, Experimento, estudo observacional, relatório de experiência, estudo de caso	PP2
Pr10	Contribuição da pesquisa	Por exemplo, Procedimento/técnica, Relatório, Ferramenta/notação, Protótipo de solução específica	PP2

Finalmente, o Pr6 focou na análise dos métodos formais empregados para melhorar a confiabilidade de sistemas críticos baseados em AM, investigando técnicas como Lógica Temporal Linear (*Linear Temporal Logics* - LTL). Esta análise demonstra como esses métodos formais contribuem para a melhoria dos sistemas críticos baseados em AM.

Para a PP2, aplicou-se várias estratégias de classificação para avaliar a natureza e o impacto dos estudos sob análise. Nos indicadores Pr7 e Pr8, compilou-se informações sobre as datas de publicação e o número de citações. Esses elementos descritivos essenciais facilitaram a identificação de tendências de pesquisa e a popularidade dos estudos ao longo do tempo. Essas informações ajudaram a detectar trabalhos amplamente referenciados ou que estavam ganhando relevância em múltiplos estudos.

Para o Pr9, uma classificação focada nos tipos de estudos empíricos permitiu uma avaliação sistemática da metodologia de pesquisa empregada nos trabalhos analisados. Finalmente, o Pr10 determinou o tipo de contribuição de pesquisa oferecida pelos estudos, categorizando-os em procedimentos ou técnicas, modelos analíticos, ferramentas ou notações e soluções específicas. Esta classificação ajudou a entender a natureza da inovação proposta e sua aplicabilidade potencial na solução de problemas práticos ou teóricos dentro do campo de estudo.

3.2 Resultados e Discussões

Esta seção explora as respostas às PPs introduzidas na seção anterior. Destacam-se os principais *insights* derivados da avaliação dos dados coletados, examinando-se seu significado no contexto da investigação.

3.2.1 Integração de Métodos Formais e AM em Ambientes Críticos

Esta seção considera os dados e a análise relacionados às PP1.1, PP1.2 e PP1.3. Posteriormente, uma análise combinada dos resultados das PP1.1, PP1.2 e PP1.3 fornece uma compreensão das dinâmicas e implicações dessa interseção no contexto crítico.

PP1.1. Quais algoritmos específicos de AM têm sido utilizados em conjunto com métodos formais?

À medida que o AM se incorpora cada vez mais em setores críticos como saúde e automação, a verificação formal ganha destaque. Essa abordagem é fundamental para garantir a confiabilidade, segurança e transparência dos modelos de AM, assegurando que eles atendam às especificações necessárias e evitem falhas ou comportamentos imprevistos. A análise dos estudos oferece *insights* cruciais sobre a integração do AM com técnicas de verificação formal. Na Figura 3.3 são detalhados os algoritmos de AM identificados. O total excede 100% porque um único estudo pode abranger múltiplos algoritmos.

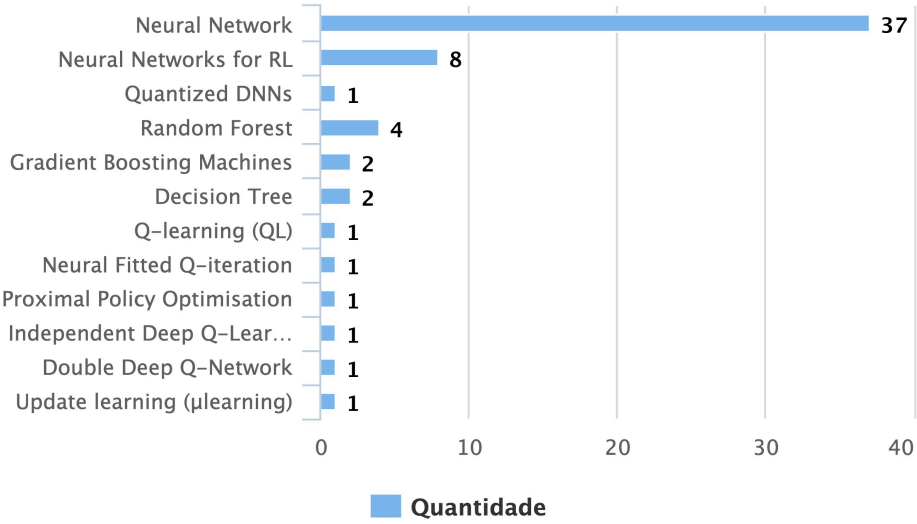

Figura 3.3: Algoritmos de AM analisados nos estudos revisados.

A revisão de vários estudos destaca uma marcada preferência pelo uso de redes neurais, mencionadas em 37 artigos analisados de várias formas (por exemplo, artificial, profunda, *feed-forward*, convolucional e recorrente). Além disso, *Quantized Deep Neural Networks"* (QDNN) foram objeto de investigação em um único artigo, enquanto redes neurais aplicadas ao aprendizado por reforço foram destacadas em oito publicações.

No campo do aprendizado por reforço, vários algoritmos específicos foram abordados individualmente em artigos distintos, incluindo *Proximal Policy Optimization* (PPO), *Independent Deep Q-Learning* (IDQL), *Double Deep Q-Network* (Double DQN), *Q-learning* (QL)

e *Neural Fitted Q-iteration* (NFQ). Além disso, um método inovador chamado aprendizado de atualização de modelo (μlearning) também foi explorado.

A diversidade de algoritmos também é evidente, onde algoritmos baseados em regras também se destacaram, com RF sendo utilizado em 4 estudos e uma combinação de RF com *Gradient Boosting Machines* emergindo em outro. Um estudo adicional investigou a interação entre DT, RF e *Gradient Boosting Machines*, refletindo um interesse crescente em explorar sinergias entre várias técnicas para aprimorar a eficácia dos sistemas de AM em cenários complexos. Esses métodos são particularmente valorizados em sistemas críticos por sua capacidade de fornecer explicabilidade, um atributo essencial para entender e confiar em decisões automatizadas.

Dos 56 artigos analisados, 45 focaram em apenas um algoritmo de AM, refletindo uma tendência à especialização e aprofundamento em técnicas específicas. Enquanto isso, 10 artigos exploraram dois tipos de algoritmos. Apenas um artigo investigou três diferentes tipos de algoritmos, demonstrando um esforço significativo para avaliar a complementaridade e eficácia de múltiplas técnicas. Esta distribuição sugere um equilíbrio entre estudos focados em profundidade e aqueles que buscam sinergias entre diferentes métodos, sublinhando a diversidade de abordagens na pesquisa de AM.

Os vários métodos destacam a complexidade e os desafios de implementar AM em ambientes sensíveis. Essa diversidade ressalta a necessidade crítica de pesquisa contínua para abordar essas dificuldades, enfatizando a importância dos avanços que promovam uma integração mais eficaz entre AM e verificação formal. Este esforço visa não apenas superar obstáculos existentes, mas também impulsionar inovações significativas na interseção desses campos.

PP1.2 Quais métodos formais têm sido empregados em conjunto com AM?

No campo do AM, a adoção de métodos formais tem se mostrado uma estratégia eficaz para garantir a correção, segurança e robustez dos modelos e algoritmos desenvolvidos. As Tabelas 3.4 e 3.5 apresentam uma análise detalhada dos métodos formais/formalismos empregados em pesquisas de AM, categorizados pelo número de artigos publicados para cada método ou formalismo. Essa abordagem quantitativa fornece *insights* sobre as tendências

atuais e a importância relativa de diferentes técnicas formais na comunidade científica de AM.

Tabela 3.4: Métodos e formalismos utilizados nos estudos revisados.

Método/Formalismo	**Qtd**
Verificação de propriedade de segurança [138, 27, 20, 47]	4
Satisfatibilidade modulo teorias [9, 59, 31]	3
Programação linear inteira mista [76, 61, 64]	3
Verificação baseada em análise de alcançabilidade [18, 118, 46]	3
Lógica temporal linear [44, 43]	2
Síntese orientada à verificação para reparo de redes neurais profundas (*Deep Neural Networks - DNNs*) [70]	1
Verificação de propriedades usando lógica de árvore computacional probabilística [78]	1
Aperto simbólico aprimorado através de otimização de erros [135]	1
Lógica temporal linear truncada [66]	1
Autômatos temporizados [69]	1
Verificação de rede neural baseada em superaproximação [40]	1
Análise de espaço de estados para verificação formal adversarial em tempo de execução [128]	1
Verificação formal usando a ferramenta Marabou [89]	1
Verificação formal baseada no Modelo Taylor [48]	1
Verificação formal baseada em programação semidefinida [34]	1
Verificação baseada em modelagem, síntese e análise usando lógica de primeira ordem extensível [19]	1
Verificação baseada em programação linear [79]	1
Verificação baseada em análise intervalar [22]	1
Abstração de estado finito, análise de alcançabilidade e satisfatibilidade modulo codificação convexa [115]	1
Verificação formal baseada em verificação de equivalência [102]	1
Verificação formal baseada em resolução de restrições usando aproximação linear [137]	1
Verificação formal baseada em propriedades de circuito fechado [73]	1
Verificação de modelo formal [134]	1

Entre os métodos formais comumente adotados, a verificação de propriedades de segurança lógica se destaca [27, 20, 138]. Esses trabalhos ressaltam a relevância dos métodos formais para o aprimoramento de sistemas críticos baseados em AM. A verificação de propriedades lógicas fornece uma base sólida para a análise de segurança, permitindo a identificação precoce de falhas e facilitando a criação de modelos mais robustos e confiáveis.

Demarchi *et al.* [27] apresentaram um método para verificar pré-condições e pós-condições expressas em fórmulas lógicas, garantindo que os modelos atendam aos critérios predefinidos de segurança e funcionalidade correta. No trabalho de Colaco e Nadjm-Tehrani [20], a verificação lógica é ampliada com o conceito de classes de equivalência, permitindo a redução do espaço de entrada por meio do agrupamento de casos semelhantes, simplificando

Tabela 3.5: Métodos e formalismos utilizados nos estudos revisados.

Método/Formalismo	**Qtd**
Abstração simbólica e consultas de robustez [103]	1
Teste de satisfatibilidade (SAT) [30]	1
Satisfatibilidade modulo convexa [114]	1
Safe-DRL estrutura de verificação formal baseada em regras de comparação intervalar de Moore [91]	1
Interpretação abstrata e alcançabilidade [75]	1
Propagação de alcance para frente baseada em QE e estrutura de verificação [96]	1
Gradiente de política lógica probabilística [49]	1
Verificação de rede neural combinando resolução SAT e programação linear [29]	1
Método para garantir a segurança do sistema e melhorar a eficiência energética [11]	1
Redes de Petri Coloridas [84]	1
Aproximação de polinômios de Bernstein, partição de estado e computação de conjunto invariante [126]	1
Verificação de propriedade LTL usando autômatos temporizados e grafos de estado [1]	1
Propriedades lógicas para verificação de modelos [1]	1
Lógica para especificar e provar propriedades de alcançabilidade de sistemas dinâmicos híbridos [45]	1
Interpretação de RF baseada em lógica de primeira ordem [71]	1
GAN condicional com uma rede de controle neural [60]	1
Verificação formal de propriedades de segurança e robustez [117]	1
Verificação formal de propriedades de segurança em tempo de execução [36]	1
Verificação formal de propriedades comportamentais com ProVe baseada em álgebra intervalar de Moore [21]	1
Verificação formal baseada em probably approximately correct [92]	1
Verificadores de plausibilidade e robustez [119]	1
Refinamento abstrato guiado por contraexemplo [26]	1
Forma normal conjuntiva [95]	1

assim o processo de verificação. Essa abordagem auxilia na identificação de inconsistências e assegura que os modelos mantenham um comportamento consistente, mesmo em situações adversas.

Enquanto isso, Zhu *et al.* [138] foca na garantia da segurança de políticas de controle aprendidas por aprendizado por reforço baseado em redes neurais. O método propõe a síntese de um programa determinístico que permite a aplicação de algoritmos de verificação formal para garantir que as políticas estejam em conformidade com as propriedades de segurança definidas para o sistema de transição de estados. Essa metodologia fornece uma camada adicional de confiança, especialmente em sistemas críticos onde a segurança é fundamental.

A *Satisfiability Modulo Theories* (SMT) também desempenha um papel importante, conforme mostrado nos estudos de Brausse *et al.* [9], Katz *et al.* [59] e Elboher *et al.* [31]. Esses estudos destacam a importância das técnicas baseadas em SMT na verificação for-

mal de modelos de AM, permitindo uma abordagem abrangente e refinada para garantir a segurança e robustez em sistemas críticos.

Braußе *et al.* [9] apresentaram um algoritmo de satisfatibilidade genérico chamado GEARSAT, com sua variante GEARSATδ para o problema SSC. Esta abordagem possibilita encontrar soluções satisfatórias para questões de segurança em redes neurais, permitindo a identificação e correção de possíveis falhas. Katz *et al.* [59] apresentaram a ferramenta Marabou como uma solução baseada em SMT para responder a consultas relacionadas às propriedades de redes neurais. Ela transforma essas consultas em problemas de satisfação de restrições, oferecendo um raciocínio de alto nível dentro da rede para reduzir o espaço de busca, melhorar o desempenho e suportar a execução paralela para aumentar a escalabilidade. Enquanto isso, Elboher *et al.* [31] propõem uma estrutura geral para superaproximação e refinamento de DNNs, juntamente com várias heurísticas de abstração e refinamento a serem usadas dentro da estrutura. Esta abordagem permite a identificação de propriedades de segurança e sua correção efetiva, tornando os modelos mais seguros e confiáveis.

Além disso, Sun *et al.* [114] introduzem a técnica de *Satisfiability Modulo Convex* (SMC), propondo um método que permite o cálculo de limites precisos sobre as probabilidades de segurança dos nós em um gráfico, mesmo diante de possíveis superestimações nas probabilidades de transição entre eles. Com base na fórmula SMC proposta, os autores desenvolveram um método heurístico para refinar a abstração do sistema, melhorando ainda mais as estimativas dos limites de segurança. O trabalho apresenta um esquema de verificação inovador para sistemas dinâmicos estocásticos com controladores de rede neural ReLU.

Elboher *et al.* [30] destacam a técnica de *Satisfiability Testing* (SAT), onde é apresentada uma melhoria na verificação baseada em abstração de redes neurais usando raciocínio residual. Este processo utiliza informações adquiridas ao verificar uma rede abstrata para acelerar a verificação de uma rede refinada, permitindo uma verificação mais rápida e eficiente.

A técnica de *Mixed Integer Linear Programming* (MILP) também se destaca, conforme apresentado nos estudos [76], [123] e [64]. Esta abordagem tem se mostrado eficaz para verificar propriedades críticas de redes neurais, fornecendo garantias formais para sistemas críticos em diversos domínios, desde sistemas de energia até a saúde.

Mistry *et al.* [76] propõem uma metodologia para a verificação de redes neurais quantizadas, abordando o problema como um modelo de decisão por meio de MILP. A técnica proposta adota uma abordagem inovadora para codificar a aritmética de ponto fixo em programas lineares inteiros mistos, oferecendo uma solução eficiente para assegurar a segurança e a precisão dessas redes.

Venzke e Chatzivasileiadis [123] se concentram na verificação do comportamento de redes neurais, assegurando formalmente a segurança em aplicações de sistemas de energia. Os autores abordam problemas de verificação, incluindo (a) provar a inexistência de exemplos adversariais, (b) avaliar a robustez das redes neurais e (c) identificar as maiores regiões de entrada com a mesma classificação. Esta abordagem oferece uma visão abrangente das possíveis falhas e vulnerabilidades das redes neurais em sistemas críticos.

Usando uma abordagem de propagação de intervalo formal, Kushner *et al.* [64] realizam a verificação de conformidade de modelos de redes neurais para dinâmicas de glicose-insulina. O objetivo é verificar se os modelos de redes neurais que prevêem os níveis futuros de glicose no sangue em indivíduos com diabetes tipo 1 são monotônicos em relação às suas entradas de insulina, garantindo a segurança e a confiabilidade desses sistemas de saúde.

A verificação de segurança baseada em análise de alcançabilidade também se destaca entre os métodos formais para a verificação de redes neurais. Conforme apresentado nos estudos de Clavière *et al.* [18], Tran *et al.* [118] e Irfan *et al.* [46], essa técnica oferece uma abordagem abrangente para demonstrar a segurança de sistemas críticos.

Clavière *et al.* [18] apresentam uma estratégia abrangente para garantir a segurança de uma classe específica de sistemas controlados por redes neurais, onde o controlador atua como um classificador composto por várias redes ReLU. Esta estratégia garante que o comportamento do sistema permaneça dentro de limites seguros, mesmo em situações desafiadoras.

Tran *et al.* [118] introduzem um algoritmo de análise de alcançabilidade exata baseado em conjuntos estelares. A técnica proposta oferece um método rápido e escalável para análise de alcançabilidade exata e superaproximada de DNNs com funções de ativação ReLU. Usando o conceito de conjunto estelar, a abordagem pode verificar com precisão as propriedades de

segurança da rede.

Finalmente, Irfan *et al.* [46] utilizam um algoritmo de análise de alcançabilidade para redes neurais em evitamento de colisão de aeronaves não tripuladas. A ferramenta de verificação Marabou garante que as redes neurais responsáveis por evitar colisões operem dentro de parâmetros seguros, prevenindo acidentes. Esses estudos ilustram a eficácia da análise de alcançabilidade na verificação de segurança de redes neurais, proporcionando garantias formais para aplicações críticas, como o controle de tráfego aéreo.

Entre os métodos formais adotados, a LTL tem sido uma escolha comum para verificar sistemas de aprendizado por reforço. Hasanbeig *et al.* [43] usaram LTL para fornecer orientação lógica que ajuda a moldar corretamente as recompensas para aprendizado por reforço. Eles convertem fórmulas LTL em um Autômato Generalizado Buchi Limitado-Determinístico, combinado com um *Markov Decision Process* (MDP), para garantir que as decisões estejam dentro dos limites de segurança.

Hasanbeig *et al.* [44] propõem uma estrutura para aprendizado por reforço seguro e verificável usando LTL. A estrutura permite que algoritmos de aprendizado por reforço sem modelo sintetizem políticas de controle para MDPs de estado finito e contínuo, garantindo que os traços gerados satisfaçam a propriedade LTL com alta probabilidade. As propriedades LTL atuam como monitores de alto nível, auxiliando no planejamento do agente e garantindo a segurança das ações.

Li *et al.* [66] introduzem a *Truncated Linear Temporal Logic* (TLTL), uma lógica temporal de predicados especificamente projetada para tarefas robóticas, e um algoritmo de aprendizado por reforço seguro guiado por autômatos. A linguagem formal de especificação permite definir explicitamente comportamentos indesejáveis, enquanto as *Control Barrier Functions* (CBFs) são usadas para garantir conformidade com as especificações de segurança. A abordagem converte fórmulas TLTL em autômatos, criando um método guiado por autômatos para aprendizado por reforço que gera funções de recompensa facilmente interpretáveis.

Para abordar o comportamento de múltiplos agentes em ambientes críticos, Mqirmi *et al.* [78] usam a *Probabilistic Computational Tree Logic* (PCTL) para verificar as propriedades

de segurança de políticas abstratas e introduzem o método de *Assured Multi-Agent Reinforcement Learning* (AMARL). Este método oferece garantias formais para o comportamento seguro de agentes operando em ambientes desconhecidos.

Por fim, Abbas *et al.* [1] usam LTL e autômatos temporizados para verificar as propriedades de sistemas de assistência em tempo real usando ferramentas como UPPAAL e IAR Visual State. Eles avaliam um sistema de assistência que detecta atividades humanas, como quedas e engasgos, demonstrando a capacidade das técnicas formais de fornecer suporte confiável para sistemas críticos.

Além de LTL, o formalismo de autômatos temporizados é usado no estudo de Lu *et al.* [69], que emprega as ferramentas DeepAuto e UPPAAL. Esses autômatos fornecem uma maneira formal de representar sistemas de controle em tempo real e analisar suas propriedades críticas, tornando-os adequados para verificar sistemas baseados em AM em cenários de segurança crítica.

Para reforçar a segurança das DNNs, Ren *et al.* [96] propõem uma estrutura de verificação baseada em eliminação de quantificadores e propagação de intervalo, conhecida como estrutura de propagação de intervalo para frente baseada em eliminação de quantificador e verificação. Esta abordagem ajuda a identificar áreas onde as redes podem ser vulneráveis a ataques adversariais e fornece garantias formais sobre os limites de segurança.

Jansen *et al.* [49] empregaram a técnica de *Probabilistic Logical Policy Gradient* (PLPG) para criar aprendizado por reforço seguro através de proteções probabilísticas. Esta técnica modela restrições de segurança lógica usando funções diferenciáveis, garantindo que as políticas aprendidas permaneçam dentro de limites seguros.

Wang *et al.* [126] apresentam um método inovador baseado em aproximação de polinômios de Bernstein, partição de estado, conversão para sistemas híbridos e computação de conjunto invariante robusto. Eles desenvolveram um método formal para analisar o espaço de configuração segura de cada controlador. A estrutura proposta garante a segurança do sistema ao longo do tempo se o estado inicial estiver dentro do espaço de configuração segura conjunta. Este espaço é calculado usando um método de aproximação inovador baseado em polinô-

mios de Bernstein. Chen *et al.* [15] verificaram e avaliaram aceleradores de redes neurais usando propriedades lógicas para verificação de modelos, focando na cibersegurança. Este método ajuda a identificar vulnerabilidades e garante que os sistemas operem de maneira segura e confiável.

No estudo de caso de Ivanov *et al.* [47], a segurança de um carro de corrida autônomo equipado com um controlador de rede neural é verificada através da técnica de verificação de propriedade de segurança. O método Verisig foca em redes neurais com funções de ativação suaves, como sigmoide ou tangente hiperbólica. O Verisig transforma a rede neural em um sistema híbrido equivalente, permitindo a aplicação de técnicas de verificação formal, como a ferramenta Flow*, para garantir a segurança das políticas de controle.

Além disso, a verificação formal baseada em classes de equivalência com verificação de plausibilidade e robustez, apresentada no estudo de Törnblom e Nadjm-Tehrani [119], propõe o uso de classes de equivalência para definir e verificar propriedades de redes neurais. Se ocorrer não conformidade, um contraexemplo é apresentado. A verificação é realizada com base em uma função representando o modelo (entrada e saída) e uma fórmula a ser verificada. As duas propriedades verificadas incluem plausibilidade de intervalo e robustez, permitindo a implementação de verificadores específicos do domínio.

Wang *et al.* [128] propõem uma abordagem para verificação de segurança adversarial em tempo de execução usando análise de intervalo e alcançabilidade. A análise de espaço de estados para verificação de segurança adversarial em tempo de execução permite identificar possíveis ataques adversariais e vulnerabilidades, proporcionando garantias formais sobre limites de segurança.

Fulton *et al.* [36] apresentaram a verificação formal de propriedades de segurança em tempo de execução para sistemas dinâmicos modelados por equações diferenciais. O método propõe modelos que combinam equações diferenciais para representar o ambiente e um programa de controle que seleciona entradas de atuadores. Além disso, inclui uma propriedade de segurança e uma prova formal de que o programa de controle restringe a dinâmica do sistema, garantindo que a propriedade de segurança nunca seja violada. A abordagem se concentra em fornecer garantias de políticas em tempo de execução por meio da verificação.

Törnblom e Nadjm-Tehrani [117] propõem um método eficiente para buscar violações contra propriedades interessantes em RF. A verificação formal de propriedades de segurança e robustez em RF usando classes de equivalência inclui segurança global e robustez contra ruído. As classes de equivalência são usadas para particionar a entrada, e o *Property Checker* verifica se todos os mapeamentos de entrada/saída capturados por cada classe de equivalência são válidos de acordo com uma propriedade lógica.

Corsi *et al.* [21] focam na verificação formal de propriedades comportamentais com o Verificador de Propriedades (ProVe) baseado na álgebra intervalar de Moore. A abordagem verifica propriedades de segurança que restringem o comportamento do agente, complementando a métrica de recompensa. Ao contrário de outros métodos, o foco está na verificação de propriedades que garantem que o agente tome decisões racionais.

A estrutura de verificação formal apresentada por Pore *et al.* [91] propõe a estrutura de verificação formal Safe-DRL baseada em regras de comparação intervalar de Moore. É definida de forma que, dada um conjunto de propriedades, a estrutura deve retornar se a propriedade é satisfeita ou fornecer contraexemplos. Os autores se baseiam nas regras de comparação intervalar de Moore para verificar a propriedade.

Hunt *et al.* [45] apresentam uma lógica para especificar e provar propriedades de alcançabilidade de sistemas dinâmicos híbridos, que combinam tanto a dinâmica discreta (por exemplo, um robô decidindo ações em tempos discretos) quanto a dinâmica contínua.

O método de *Counterexample-Guided Abstract Refinement* (CEGAR), proposto por Demarchi e Guidotti [26], guia a abstração e o refinamento para a verificação de redes neurais. A técnica gera contraexemplos que orientam o processo de refinamento, tornando a verificação mais precisa e eficiente. Guidotti [40] apresenta um algoritmo de verificação baseado em superaproximação para verificar redes neurais. O método usa uma abordagem de superaproximação para identificar possíveis violações de propriedades de segurança e robustez.

O estudo de Ma *et al.* [71] aborda a interpretação de RF. Os autores propõem a interpretação de RF baseada em lógica de primeira ordem. Eles codificam o processo de tomada de decisão dessas florestas em fórmulas de lógica de primeira ordem, permitindo uma melhor

interpretabilidade das decisões.

Para garantir segurança e eficiência energética, Bunel *et al.* [11] introduzem um método de comutação automática entre vários controladores para garantir a segurança do sistema e melhorar a eficiência energética. O algoritmo Double DQN é usado para aprender uma estratégia adaptativa eficiente em termos de energia com garantias de segurança, formulando o processo de aprendizagem como um MDP.

A verificação de robustez de DNNs com ativação LeakyReLU é explorada no estudo de Mellouki *et al.* [75]. A abordagem de verificação de robustez de DNNs com LeakyReLU através de interpretação abstrata e alcançabilidade cai sob as abordagens de alcançabilidade. O ETH *Robustness Analyzer for Neural Networks* (ERAN) verifica a robustez das redes contra perturbações de entrada convertendo as camadas da rede em camadas abstratas. Zelazny *et al.* [135] apresentam a verificação de DNNs com aperto simbólico aprimorado através de otimização de erros (DeepMIP), que visa otimizar métodos de substituição de back-substitution para verificação de redes neurais.

A verificação de segurança baseada em modelagem, síntese e análise usando lógica de primeira ordem extensível, juntamente com *Runtime Assurance* (RTA), é apresentada no estudo de Cofer *et al.* [19]. O RTA é aplicado junto com ferramentas de síntese formal, modelagem e análise a um sistema de prevenção de colisões aéreas baseado em redes neurais. A arquitetura RTA garante a segurança de um sistema autônomo de táxi de aeronaves, fornecendo garantias formais sobre suas propriedades de segurança.

Prashant e Easwaran [92] apresentam a verificação formal baseada em PAC para detecção de dados *Out-of-Distribution (OOD)*, visando criar uma estrutura para garantir e limitar falhas na detecção de OOD. O procedimento de verificação de segurança verifica se as instâncias de dados amostradas estão dentro de uma região segura do hiper-espaço codificado.

A estrutura NNLander-VeriF, introduzida no estudo de Santa Cruz e Shoukry [103], foca na verificação de segurança e *liveness* baseada em abstração simbólica e consultas de robustez de redes neurais. A abstração simbólica da dinâmica física da aeronave divide o problema de verificação de modelos de propriedades de segurança e *liveness* em um conjunto de consultas

de robustez de redes neurais.

Sun *et al.* [115] apresentaram um método para verificação de segurança baseado em abstração de estado finito, análise de alcançabilidade e codificação SMC. O método pode raciocinar sobre a segurança do sistema considerando a dinâmica contínua do robô, a configuração do espaço de trabalho, a sensoriamento LiDAR e a rede neural.

A abordagem de verificação de redes neurais combinando resolução de *Satisfiability* (SAT) e programação linear com aproximação linear do comportamento da rede é proposta no estudo de Ehlers [29]. O método combina resolução SAT com programação linear, usando uma nova aproximação linear do comportamento da rede.

Paterson *et al.* [89] integraram a ferramenta DeepCert com a ferramenta Marabou para codificar perturbações contextuais. A estrutura completa é capaz de analisar redes neurais usando funções de perturbação de entrada.

A técnica apresentada no estudo de Katz *et al.* [60] integra um *Conditional GAN* (cGAN) com uma rede neural de controle para caracterizar o conjunto de entrada plausível para um controlador neural baseado em imagem. A técnica se baseia no treinamento de um cGAN para produzir imagens realistas para um dado estado e concatenar a rede geradora com a rede de controle. Esta integração reduz o problema de verificação para verificar um controlador neural baseado em estado, permitindo o uso de técnicas existentes.

Raman *et al.* [95] apresentam a técnica de *Conjunctive Normal Form* (CNF). CNF é uma conjunção de uma ou mais cláusulas, cada uma delas uma disjunção de variáveis, e desempenha um papel crucial na verificação formal. Zhang *et al.* [137] propõem uma abordagem de verificação formal baseada na resolução de restrições usando aproximação linear. A técnica sugere converter o problema de verificação de robustez em um problema de resolução de restrições através de aproximação linear. Além disso, apresentam um método específico para verificar a robustez de redes neurais usando a função de ativação não monótona Swish.

A verificação de DNNs em *hardware* é abordada por Saji *et al.* [102]. A abordagem de verificação formal baseada na verificação de equivalência cria uma infraestrutura de verificação leve para verificar a lógica de DNNs através da verificação de equivalência.

Corsi *et al.* [22] discutem a verificação formal baseada em análise intervalar. A abordagem sugere usar análise intervalar para verificar as relações entre duas ou mais saídas de rede, permitindo a verificação de modelos de aprendizado por reforço profundo nos quais os nós de saída correspondem a ações e a DNN é responsável por selecionar a melhor ação.

Ma *et al.* [70] abordam a técnica de síntese orientada à verificação para reparo de DNNs. O método propõe uma estrutura inovadora para corrigir violações de propriedades de segurança e ataques de backdoor em DNNs.

Muller *et al.* [79] apresentam a técnica PRIMA, que oferece verificação formal baseada em programação linear. A abordagem PRIMA é baseada em otimização e pode lidar com qualquer especificação de segurança, seja pré-condições ou pós-condições, desde que sejam expressas como poliedros convexos. Usar um solucionador de programação linear permite encontrar limites precisos para a certificação de redes neurais.

A técnica de verificação formal baseada em programação semidefinida é introduzida por Fazlyab *et al.* [34]. A abordagem desenvolve uma nova estrutura baseada em programação semidefinida para verificar a segurança e analisar a robustez de redes neurais contra distúrbios de entrada limitados por normas. O problema de viabilidade *linear matrix inequality* é suficiente para verificar a segurança da rede neural.

No estudo de Lopez *et al.* [73], é proposta a verificação formal baseada em propriedades de circuito fechado. O método estende o benchmark de sistema fechado para incluir 10 propriedades que avaliam a segurança de uma aeronave na presença de outra aeronave intrusa na mesma altitude.

Ivanov *et al.* [48] apresentam a verificação formal baseada no modelo de Taylor. A abordagem é mais escalável, adotando o modelo de Taylor em vez da integração da dinâmica sigmoide usada no método Verisig.

Yel *et al.* [134] introduzem a técnica de verificação de modelo formal. A abordagem propõe verificação rápida de segurança e planejamento para operações de veículos autônomos em ambientes congestionados sob distúrbios de tempo de execução desconhecidos. Finalmente, Nauman *et al.* [84] propuseram um método baseado em redes de Petri coloridas (*Coloured*

Petri Nets - CPN), visando melhorar a precisão dos modelos de DT para diagnósticos médicos. A abordagem usa CPN para garantir a correção das regras de decisão e eliminar regras duplicadas.

Assim, a aplicação de métodos formais no contexto de AM é extensa e variada, abrangendo a verificação de propriedades lógicas e temporais e a análise da robustez e segurança de modelos complexos. Cada método contribui para garantir a confiabilidade e segurança dos sistemas de AM. A inovação contínua e a adaptação dessas técnicas formais são essenciais para enfrentar os desafios emergentes no domínio do AM, garantindo o desenvolvimento de sistemas confiáveis, seguros e dignos de confiança.

PP1.3. Quais métodos e ferramentas formais foram identificados?

Na Tabela 3.6 são apresentadas ferramentas específicas que facilitam a implementação dos métodos e formalismos destacados anteriormente. Ferramentas para verificação de conjuntos de árvores, RF e DNNs destacam o foco crescente na verificação de segurança. Alguns estudos implementaram novas ferramentas baseadas em métodos propostos usando linguagens de programação como Python. Estes são omitidos da Tabela 3.6 porque não é fornecido nenhum nome de ferramenta e descrição detalhada.

Ferramentas como Marabou, DeepCert, ERAN e NEVER2 são fundamentais na verificação de DNNs, fornecendo análises abrangentes e precisas. Também é importante destacar a integração de ferramentas como DeepCert com Marabou, permitindo uma codificação eficaz de perturbações contextuais e uma análise completa das redes neurais.

O provador de teoremas ACL2, por exemplo, oferece uma abordagem poderosa para garantir a segurança de redes neurais por meio da prova formal de teoremas. Da mesma forma, o provador de teoremas KeYmaera X e a ferramenta UPPAAL são cruciais na verificação de propriedades temporais e lógicas.

O ERAN é outra ferramenta usada para verificar a robustez de DNNs contra perturbações de entrada. A ferramenta NEVER2 suporta a análise de intervalos de saída usando abstração politópica, oferecendo uma abordagem inovadora para a verificação formal de redes neurais.

Ferramentas como o solucionador de restrições gurob, o solucionador Gurobi MILP e o solucionador Z3 são essenciais na resolução de problemas de programação linear e lógica, fornecendo a base para várias abordagens de verificação. JasperGold desempenha um papel fundamental na verificação de propriedades lógicas, enquanto o verificador de modelos Storm verifica propriedades probabilísticas. As ferramentas *Verifier of Random Forests* (VoRF) e *Verifier of Tree Ensembles* (VoTE) são dedicadas à verificação de modelos de AM baseados em árvores e florestas, respectivamente, fornecendo uma análise robusta desses modelos.

Tabela 3.6: Ferramentas utilizadas pelos estudos revisados.

Nome da Ferramenta	**Qtd**
ACL2 theorem prover [19]	1
CBMC [95]	1
CPN Tools [84]	1
Constraint solver gurob [137]	1
DeepAuto e UPPAAL [69]	1
DeepCert [89]	1
ERAN [75]	1
Extend the state-of-the-art Marabou DNN verification engine [30]	1
Gurobi MILP solver [76]	1
JasperGold [15]	1
KeYmaera X theorem prover [36]	1
Marabou tool [59, 46]	2
Modified DEEPZ verification algorithm [60]	1
NEVER2 [26, 27]	2
NNLander-VeriF [103]	1
NNV tool [73]	1
ProVe [21]	1
pyNeVer [40]	1
SMT solver [71]	1
SMC solver [115]	1
Storm model checker [78]	1
MATLAB, Mosek 8, and Yalmip [11]	1
VSRL system and KeYmaera X theorem prover [45]	1
VoRF [117]	1
VoTE Tool [119, 20]	2
Z3 solver [9]	1

Embora na Tabela 3.6 sejam resumas as ferramentas que podem ser importantes para a melhoria dos sistemas críticos baseados em AM, 11 artigos não especificaram o nome da ferramenta utilizada. Isso sugere a existência de outras ferramentas ou metodologias ainda não

claramente definidas, o que contribui para a diversidade de abordagens no campo. Portanto, a aplicação de métodos formais com o uso de ferramentas destaca a necessidade de metodologias e ferramentas robustas para garantir a segurança e precisão de sistemas críticos baseados em AM.

Juntas, as Tabelas 3.4 e 3.6 refletem o cenário atual da pesquisa e prática em verificação formal, mostrando um equilíbrio entre teoria (métodos/formalismos) e aplicação (ferramentas). Isso destaca a importância crítica da verificação formal e da análise de segurança em sistemas críticos contemporâneos e sublinha a necessidade imperativa de desenvolvimento contínuo e adaptação de novos métodos e ferramentas para enfrentar desafios emergentes em tecnologias de ponta, que vão desde AM até sistemas ciberfísicos.

PP1.4. Em quais contextos críticos a combinação de métodos formais e AM foi aplicada?

A análise dos artigos revela a aplicação da combinação de métodos formais e AM em diversos contextos críticos. Assim, 25% dos artigos abordam veículos autônomos, destacando a importância do desenvolvimento da autonomia e precisão no controle. Esse interesse reflete um compromisso com o avanço das capacidades de navegação e operação independente, essenciais para a evolução e segurança dessas tecnologias.

A prevenção de colisões de aeronaves não tripuladas representa 25% dos artigos, enfatizando a segurança no tráfego aéreo e nas aeronaves autônomas. Isso sublinha a importância de soluções tecnológicas avançadas para mitigar riscos e melhorar a segurança operacional nesses domínios críticos.

Sistemas gerais de segurança crítica abrangem 23,2% dos artigos e cobrem várias aplicações. Isso destaca a necessidade imperativa de melhorar a confiabilidade e mostra um esforço contínuo para garantir a integridade em setores onde falhas podem ter consequências significativas.

Os artigos que abordam robótica representam 8,9%, refletindo a importância do uso de robôs para auxiliar em várias tarefas. Enquanto isso, a assistência médica é o foco de 7,1% dos artigos, enfatizando a relevância de metodologias de suporte à tomada de decisão em contextos

críticos, como cirurgias robóticas.

Sistemas de controle e aeronaves autônomas representam cada um 3,6% dos artigos, destacando a importância da precisão e eficiência nos sistemas de controle e na operação de aeronaves autônomas. Finalmente, carros de corrida autônomos e controle de sistemas de energia representam cada um 1,8% dos artigos, destacando a importância de garantir segurança e precisão em contextos mais especializados.

A diversidade de aplicações de métodos formais e AM em contextos críticos sublinha a importância fundamental dessa integração, não apenas para impulsionar inovações tecnológicas, mas também para garantir que esses avanços sejam seguros e confiáveis. Essa convergência é crucial para lidar com as complexidades e incertezas inerentes aos sistemas críticos, promovendo uma evolução tecnológica que é tanto avançada quanto cautelosa. A adoção dessas metodologias em áreas como aeronáutica, robótica e assistência médica demonstra um compromisso com a excelência operacional e a minimização de riscos, abrindo caminho para um futuro onde a tecnologia avança de forma responsável, protegendo e melhorando vidas humanas.

Na Figura 3.4 são mapeiados os principais contextos críticos onde as técnicas de AM foram aplicadas, destacando algoritmos específicos e o número de artigos que abordam cada combinação. As interseções onde essas bolhas se sobrepõem mostram os números internos que representam o volume de pesquisas combinando a aplicação em contextos críticos com algoritmos específicos de AM. Números maiores indicam uma colaboração mais substancial entre esses campos, potencialmente sugerindo um nível mais alto de desenvolvimento na integração de métodos formais com técnicas específicas de AM. Com um foco significativo em redes neurais e técnicas de aprendizado por reforço, os estudos cobrem muitas aplicações práticas, destacando a evolução e o potencial impacto dessas tecnologias.

Nas interseções onde essas bolhas da Figura 3.4 se sobrepõem, os números internos representam o volume de pesquisas que consideram algoritmos de AM (verificados usando métodos formais) e cenários de aplicação crítica. Números maiores indicam uma colaboração mais substancial entre esses campos, potencialmente indicando um nível mais alto de desenvolvimento na integração de métodos formais com técnicas particulares de AM em um contexto

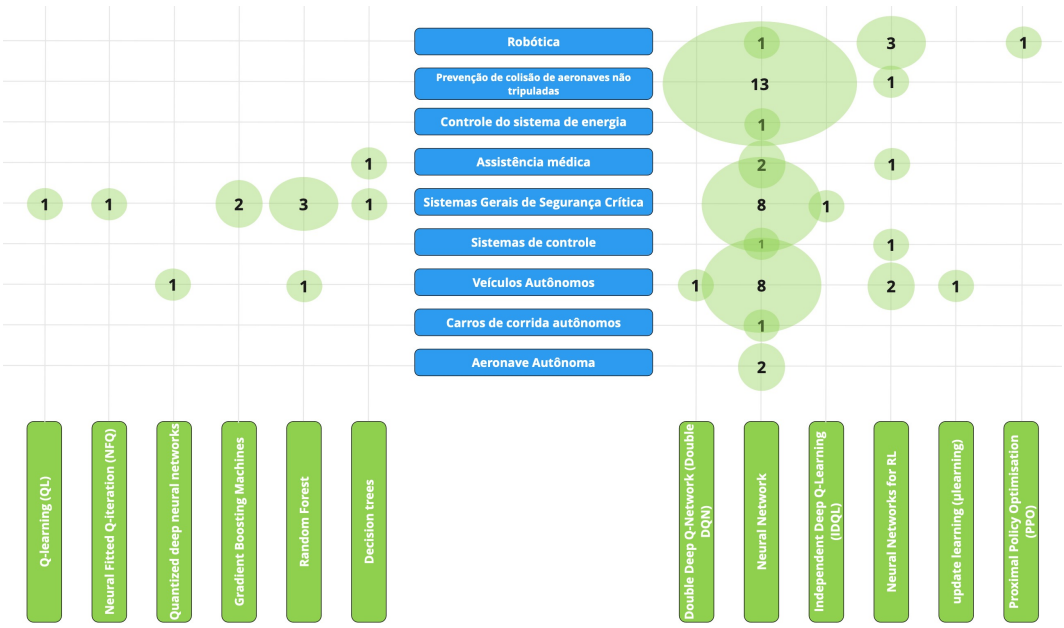

Figura 3.4: Mapeamento de contextos críticos e algoritmos de AM identificados.

específico. Essa tendência reflete uma inclinação crescente dentro da comunidade científica em buscar validação formal de modelos de AM para garantir confiabilidade, segurança e correção. No entanto, também mostra estudos de caso limitados focando em sistemas críticos. Por exemplo, 8 artigos focam em sistemas de segurança crítica geral. Portanto, esses 8 estudos apenas discutem o contexto sem conduzir estudos de caso específicos.

Assim, o diagrama visualiza a pesquisa interdisciplinar em andamento, oferecendo *insights* sobre como os métodos formais evoluem para acomodar o rápido progresso nos algoritmos de AM em vários contextos. Há um foco crescente em refinar técnicas sofisticadas de AM, garantindo simultaneamente sua verificação e validação rigorosas para implantação em cenários críticos. Essa integração de abordagens está promovendo melhorias significativas na confiabilidade e aplicabilidade do AM, abrindo caminhos para seu uso em contextos críticos.

Consequentemente, o estado atual da arte na convergência desses domínios reflete um esforço concertado para enfrentar as complexidades e desafios de integrar AM em ambientes intolerantes a falhas. Essa busca colaborativa fomenta o progresso que combina inovação tecnológica com rigor matemático. Tal abordagem multidisciplinar não só amplia os limi-

tes da compreensão científica e tecnológica, mas também garante que as vantagens do AM possam ser aproveitadas de forma segura e eficiente em uma ampla gama de aplicações.

As redes neurais são amplamente utilizadas em contextos críticos, como prevenção de colisões de aeronaves não tripuladas, com 13 artigos dedicados a esse tópico, enfatizando a importância crítica da segurança em sistemas aéreos autônomos. Além disso, veículos autônomos e sistemas de segurança crítica geral também recebem atenção considerável, com oito artigos cada um explorando diferentes aspectos desses sistemas e ilustrando a necessidade de desenvolver tecnologias que garantam operações seguras e eficientes. Outros setores, como saúde, sistemas de controle e controle de aeronaves, também se beneficiam das inovações na aplicação de métodos formais para aumentar a confiança nas redes neurais.

Nos sistemas de segurança crítica geral, propriedades específicas de segurança foram discutidas para garantir a confiabilidade e robustez das operações. Os estudos revisados focam em diferentes abordagens, como redes neurais, DT e RF. Nos veículos autônomos, há uma forte presença de técnicas de AM visando otimizar a navegação, prevenir colisões e melhorar a tomada de decisão, com seis tipos distintos, como redes neurais, redes neurais para aprendizado por reforço e RF. A análise destaca a aplicação diversa de técnicas de AM em contextos críticos.

PP1. Qual é o estado da arte na interseção entre métodos formais e AM?

A análise de aplicações específicas de AM (PP1.1) demonstra uma ampla implementação de algoritmos, incluindo redes neurais, aprendizado por reforço, DT e métodos de *ensemble* (como RF e *gradient boosting*). A diversidade desses métodos evidencia a flexibilidade e a eficiência do AM em diferentes cenários, destacando especialmente o papel das redes neurais devido à sua capacidade de lidar com padrões complexos em grandes conjuntos de dados.

A abordagem dos métodos formais (PP1.2) abrange uma variedade de técnicas para verificar e validar sistemas de AM, incluindo LTL, SMT e ferramentas específicas como UPPAAL. Essas técnicas garantem a correção, segurança e robustez dos modelos de AM, com atenção especial à verificação de propriedades temporais e lógicas. A introdução de ferramentas dedicadas para a verificação de modelos complexos (PP1.3), como DNNs, indica um foco

crescente em garantir a correção dessas tecnologias avançadas.

A integração dessas técnicas em contextos críticos (PP1.4) demonstra um esforço significativo para complementar AM e métodos formais, especialmente em domínios que requerem alta confiabilidade, como sistemas autônomos e prevenção de colisões aéreas. Essa convergência visa superar as limitações inerentes aos sistemas críticos baseados em AM, introduzindo rigor formal na verificação de modelos para garantir que operem dentro de parâmetros seguros e previsíveis.

Considerando as respostas dos PPs anteriores, foi possível fornecer uma visão geral do estado da arte na interseção entre métodos formais e AM, mostrando um campo dinâmico e em evolução. Ilustra como diferentes algoritmos de AM e técnicas de métodos formais são frequentemente empregados. Vários algoritmos de AM são abordados, incluindo redes neurais, DT e RF. Isso reflete a diversidade das abordagens de AM exploradas. Técnicas formais, como LTL, destacam a amplitude dos métodos formais usados com AM.

A verificação formal em tempo de execução de políticas de aprendizado por reforço é um exemplo de uma abordagem relevante para garantir a segurança de sistemas críticos, baseando-se em fórmulas LTL e modelos de autômatos. Além disso, a verificação formal dos resultados desejados em relação aos limites de segurança é outra abordagem relevante aplicável a diversos modelos de AM. Essa abordagem pode se basear em classes equivalentes para reduzir a faixa de possíveis entradas, mantendo a confiança no procedimento de verificação.

3.2.2 Avaliação da Maturidade na Interseção de Métodos Formais e AM em Contextos Críticos

Esta seção fornece uma análise detalhada da progressão e significância das metodologias que combinam métodos formais e AM em ambientes críticos, abordando as questões de pesquisa PP2.1 e PP2.2. Subsequentemente, sintetizou-se os *insights* derivados das respostas a essas três PPs para abordar a PP2.

PP2.1. Qual é o impacto da evolução das publicações e citações em artigos científicos em AM e métodos formais em contextos críticos, e como esses fatores se correlacionam?

Compreender a distribuição temporal das publicações é essencial para avaliar a evolução e relevância de um campo de pesquisa. Uma análise dos 56 artigos selecionados revelou uma diversidade de publicações ao longo dos anos, conforme apresentado na Tabela 3.8.

Tabela 3.7: Evolução das publicações e citações.

Ano	Total de Artigos Publicados	Total de Citações
2017	1	777
2018	1	374
2019	8	1203
2020	14	931
2021	10	183
2022	15	211
2023	6	116
2024	1	0
Total	56	3795

Analisando a frequência de publicação, nota-se um crescimento significativo a partir de 2019, com oito publicações. Esse aumento continuou em 2020, com nove artigos, e em 2021, com oito artigos, atingindo um pico de 15 publicações em 2022 antes de cair para seis em 2023. Essa tendência sugere um interesse crescente e a expansão do campo nos últimos anos, especialmente com o pico de publicações em 2022, que pode refletir os avanços tecnológicos rápidos e a necessidade de abordagens confiáveis em sistemas críticos baseados em AM. A análise temporal fornece *insights* valiosos sobre a dinâmica da área de pesquisa e sua importância crescente.

Compreender o impacto e a visibilidade dos artigos científicos é igualmente crucial para avaliar o progresso e a relevância de um campo de pesquisa. Portanto, analisar os números de citações é fundamental, oferecendo *insights* sobre quais artigos receberam um maior reconhecimento ao longo do tempo.

Os dados apresentados na Tabela 3.8 fornecem uma perspectiva abrangente sobre o desempenho e impacto dos artigos ao longo do tempo, destacando a relevância acadêmica e o impacto dos estudos sobre AM e métodos formais em contextos críticos. Com 3.795 citações, esses

dados refletem a frequência com que os trabalhos foram referenciados em outros estudos, indicando sua qualidade e influência dentro da comunidade científica.

As citações são essenciais para avaliar a importância e a autoridade de um artigo, pois indicam o reconhecimento de outros pesquisadores sobre o valor do trabalho citado em suas próprias pesquisas. A distribuição das citações revela padrões notáveis:

- Em 2017 e 2018, apesar de apenas um artigo ter sido publicado em cada ano, as citações foram excepcionalmente altas (777 e 374, respectivamente), sugerindo que esses artigos abordam temas pioneiros ou fundamentais no campo.
- O ano de 2019 representou um aumento nas publicações (8 artigos) e citações (1.203), marcando um período de contribuições substanciais de pesquisa.
- Em 2020, houve 14 publicações, com uma redução nas citações para 931, enquanto em 2022, houve o maior volume anual de publicações, com 15 artigos, embora as citações tenham totalizado apenas 211.

A abordagem atual é crucial para entender o panorama científico na área estudada. Sugere que, à medida que o campo de AM e métodos formais evolui, o impacto e a influência dos trabalhos publicados variam ao longo do tempo. O número total de citações serve como uma métrica robusta para entender o legado e a relevância dos estudos realizados, oferecendo *insights* sobre a evolução das tendências e metodologias.

PP2.2 Quais contribuições de pesquisa os estudos sobre a convergência de métodos formais e AM forneceram?

A pesquisa sobre a convergência de métodos formais e AM forneceu contribuições valiosas, incluindo o desenvolvimento de ferramentas especializadas, técnicas inovadoras, métodos formais adaptados e estudos de caso demonstrando aplicações práticas em contextos críticos.

Várias ferramentas foram desenvolvidas ou reutilizadas para facilitar a implementação de métodos e formalismos, incluindo DeepAuto e UPPAAL para verificação de sistemas de aprendizado profundo, VoTE e VoRF para verificação de DT e ERAN para análise de robustez de DNNs. Ferramentas como Marabou, DeepCert, NEVER2, NNLander-VeriF e ProVe

são cruciais na verificação de modelos de aprendizado profundo e fornecem análises abrangentes e precisas.

Técnicas inovadoras, como Interpretação Abstrata, CEGAR e *Symbolic Bound Tightening* (DeepMIP), foram desenvolvidas para melhorar a precisão e eficiência da verificação de redes neurais. Métodos formais como LTL, SMT e MILP foram adaptados para verificar modelos de AM.

Estudos de caso ilustram aplicações práticas em contextos críticos, como veículos autônomos e prevenção de colisões, onde métodos formais são usados para garantir a segurança de veículos e sistemas de prevenção de acidentes. Em robótica e cirurgia robótica, métodos formais ajudam a verificar algoritmos de controle; na saúde, são usados para verificar sistemas de suporte à decisão.

Portanto, a convergência entre métodos formais e AM forneceu contribuições significativas, incluindo ferramentas inovadoras, técnicas de verificação precisas e métodos formais adaptados para enfrentar desafios únicos em sistemas baseados em AM. A variedade de estudos de caso e aplicações práticas ilustra a importância dessa integração para garantir segurança e confiabilidade em sistemas críticos, promovendo uma evolução tecnológica responsável que protege e melhora a vida humana.

PP2. Qual é o nível de maturidade das soluções existentes na interseção de métodos formais e AM em contextos críticos?

Ao longo dos anos, a análise das publicações e citações revela um campo crescente e em constante evolução na interseção entre métodos formais e AM em contextos críticos. Desde 2019, foi observado um aumento significativo nas publicações, atingindo um pico de 15 artigos em 2022, sugerindo um interesse crescente e expansão nessa área nos últimos anos. A análise dos dados de citação também revela padrões importantes sobre a influência e impacto da pesquisa ao longo do tempo. Em 2019, houve um aumento significativo no número de publicações (8 artigos) e citações (1.203), marcando um período de contribuições substanciais de pesquisa. No entanto, o crescimento das citações diminuiu nos anos subsequentes, mesmo com mais publicações.

A pesquisa sobre a convergência entre métodos formais e AM forneceu contribuições, como o desenvolvimento de ferramentas especializadas, técnicas inovadoras e métodos formais adaptados. Ferramentas como DeepAuto e UPPAAL, VoTE, VoRF, ERAN, Marabou e ProVe são cruciais na verificação de modelos de aprendizado profundo e fornecem análises abrangentes e precisas. Técnicas inovadoras, como o refinamento abstrato guiado por contraexemplo e a verificação de DNNs com reforço simbólico de limites aprimorado por otimização de erro, foram desenvolvidas para melhorar a precisão e eficiência da verificação de redes neurais. Métodos formais como LTL, SMT e MILP foram adaptados para verificar modelos de AM.

Estudos de caso revelam como essas contribuições são aplicadas em cenários críticos, incluindo veículos autônomos e prevenção de colisões, robótica e sistemas de suporte à decisão na área da saúde. A variedade de casos reais e ferramentas desenvolvidas ressalta o papel crucial da combinação de métodos formais e AM para garantir segurança e confiabilidade em sistemas críticos.

Portanto, embora o campo de pesquisa esteja crescendo, o nível de maturidade das ferramentas propostas ainda parece estar em um estágio inicial. Em termos de escalabilidade, as soluções existentes podem ser criticadas, pois não são maduras o suficiente para aplicações no mundo real e geralmente são difíceis de escalar para sistemas grandes. Muitas propostas, por simplicidade, são avaliadas usando conjuntos de dados comuns como MNIST, resultando em apresentações de validação preliminares. A diversidade das contribuições e métodos de pesquisa sinaliza um campo interdisciplinar com potencial significativo para impactar positivamente contextos críticos, mas que ainda requer mais desenvolvimento e consolidação para alcançar plena maturidade e aplicabilidade prática.

3.3 Análise Comparativa

Nesta seção, realiza-se uma análise comparativa entre os principais trabalhos relacionados e o método desenvolvido nesta tese. A Tabela **??** apresenta algumas das principais abordagens identificadas na literatura, destacando os contextos de aplicação, as técnicas empregadas e as limitações observadas. Essa comparação permite evidenciar as contribuições da abordagem

proposta e demonstrar suas vantagens em relação às soluções existentes.

A maioria dos estudos revisados concentra-se na aplicação de métodos formais para validar modelos de AM em sistemas críticos. No entanto, muitas dessas abordagens apresentam limitações em termos de escalabilidade, explicabilidade e aplicabilidade prática. Alguns trabalhos focam exclusivamente na verificação formal de redes neurais profundas, enquanto outros exploram técnicas como aprendizado por reforço, ou modelos baseados em DT, sem garantir a explicabilidade do modelo resultante.

Tabela 3.8: Comparação entre os principais trabalhos relacionados e o método apresentado

Trabalho	**Técnica Utilizada**	**Limitações**
Muhammad et al. (2021)	Modelagem de DT com Redes de Petri Coloridas (CPN)	Foca na simulação do comportamento da rede, modelagem e ajustes manuais.
Törnblom e Nadjm-Tehrani (2019)	Verificação formal de RF (VoRF)	Verifica propriedades de segurança, mas não trata a eliminação de regras redundantes.
Törnblom e Nadjm-Tehrani (2020)	Verificação formal de Tree Ensembles (VoTE)	Aplica-se a RF e boosting, mas não modela regras explicitamente.
Colaco e Nadjm-Tehrani (2023)	Extensão do VoTE para verificação de estabilidade de modelos	Foca na estabilidade contra perturbações geométricas, sem eliminação de regras inconsistentes.
Nosso Método	Modelagem e ajustes automatizados de regras DT e RF em CPN	Alto custo computacional para o processamento de modelos de grande porte.

Observa-se que, enquanto métodos como VoRF e VoTE se concentram na verificação de propriedades de segurança e estabilidade de modelos baseados em DT, eles não abordam diretamente a modelagem e análise formal das regras de decisão. Além disso, o estudo de Nauman *et al.* [86] emprega CPN, propondo sua utilização para garantir que as regras de prognóstico derivadas de DT sejam verificadas. No entanto, o modelo CPN é gerado manualmente, sem um processo automatizado de extração e refinamento das regras. Nesse método, as regras foram avaliadas individualmente por meio de simulação e análise do espaço de estados para calcular sua precisão. Caso a classificação produzida pelo modelo coincidisse com o rótulo original, a regra era considerada correta. Caso contrário, era necessário um ajuste, realizado através da modificação dos atributos da regra, aumentando ou diminuindo cada valor em incrementos de 10 unidades. Esse processo era repetido iterativamente até que todas as regras fossem ajustadas para permitir a passagem do número máximo de *tokens* pelo modelo CPN.

Entretanto, a abordagem de Nauman *et al.* [86] apresenta limitações significativas, pois

não contempla a remoção de regras redundantes, a geração automatizada do modelo CPN nem a identificação de regras enganosas. O processo de modelagem é manual, tornando a abordagem menos escalável e mais suscetível a erros humanos. Além disso, a ausência de um mecanismo automatizado para detectar e corrigir regras inconsistentes pode comprometer a confiabilidade do sistema, resultando em classificações imprecisas e na necessidade de ajustes manuais demorados.

Em contrapartida, o método apresentado nesta tese incorpora a modelagem automática de regras em CPN, permitindo a simulação das regras de decisão extraídas de modelos de DT e RF. Esse processo possibilita a identificação de redundâncias, inconsistências e regras enganosas, garantindo maior explicabilidade e confiabilidade nas decisões tomadas pelo modelo.

Apesar da maioria dos artigos terem sido utilizado redes neurais, a escolha por DT e RF como base para os modelos de AM justifica-se pela necessidade de equilibrar explicabilidade e acurácia. As DTs oferecem regras simples e interpretáveis, tornando-se fundamentais para garantir transparência em sistemas críticos. Por outro lado, o RF melhora a confiabilidade e a acurácia da classificação ao combinar múltiplas árvores.

Diferentemente de outras técnicas formais, como LTL, as CPNs oferecem suporte para modelagem e análise de sistemas complexos que envolvem concorrência, paralelismo e múltiplos estados alcançáveis. Tais propriedades tornam-se essenciais para a validação de modelos em sistemas críticos, onde a robustez e a confiabilidade são fatores determinantes.

Ao combinar CPN com DT e RF, o método apresentado possibilita preencher lacunas importantes nos trabalhos existentes. Ele não apenas garante a confiabilidade do modelo por meio da validação formal, mas também promove a transparência, facilitando a compreensão e revisão das regras de decisão. Além disso, a eliminação de redundâncias e a identificação de regras problemáticas contribuem para a redução de riscos associados a decisões equivocadas.

3.4 Implicações

Com base nas análises e discussões apresentadas, as implicações para a comunidade científica e especialistas da indústria são significativas e variadas. Esta seção detalha tais im-

plicações, enfatizando o papel vital dos achados na fusão de técnicas formais com AM em contextos críticos e na avaliação do nível de desenvolvimento dessas abordagens. Em resumo, este estudo destaca a importância da colaboração estreita entre academia e indústria para impulsionar a aplicação de métodos formais e AM em ambientes críticos. O desenvolvimento contínuo de tecnologias, ferramentas e metodologias atenderá às necessidades atuais e abrirá caminho para inovações futuras que promovam sistemas mais seguros e confiáveis. O desenvolvimento de ferramentas que não exijam conhecimento prévio em métodos formais por parte do usuário parece ser uma direção promissora para a aceitação e viabilidade prática dessas soluções.

3.4.1 Implicações para Pesquisadores

A união entre métodos formais e AM em contextos críticos revela uma área de pesquisa promissora. Assim, algumas implicações-chave para pesquisadores incluem:

- A predominância das redes neurais na pesquisa indica um foco significativo nessa abordagem, destacando a necessidade de avaliar outros algoritmos de AM para aferir sua eficácia e segurança. Por exemplo, métodos baseados em DT são amplamente aceitos e adotados em aplicações médicas por gerarem modelos interpretáveis. A tendência de concentrar-se em algoritmos específicos, em vez de integrar diferentes técnicas, sugere uma divisão no campo entre análises especializadas e esforços para estabelecer conexões entre áreas diversas. Isso sinaliza inúmeras oportunidades para abrir novos caminhos e inovações por meio da fusão de metodologias variadas de AM com técnicas de verificação formal.

- O uso de métodos formais e AM reforça a necessidade de metodologias rigorosas para garantir a segurança, precisão e resiliência dos sistemas de AM. O conjunto de técnicas, como LTL e SMT, ilustra um domínio em expansão, propício à inovação. Os achados indicam um esforço concentrado para abordar complexidades específicas na verificação de modelos de aprendizado profundo, incentivando a criação de novas ferramentas e metodologias. Isso aponta para um domínio de pesquisa relevante, com grande potencial para contribuições significativas.

- A pesquisa destaca a importância de desenvolver novas técnicas e procedimentos adaptados para lidar com os desafios da integração de AM em ambientes críticos. O foco na inovação de métodos de verificação e validação abre caminhos para explorar abordagens inéditas que garantam a segurança, confiabilidade e resiliência de sistemas críticos baseados em AM.

- A necessidade de ferramentas e notações dedicadas para simplificar a aplicação de métodos formais no AM indica uma promissora linha de pesquisa futura. O desenvolvimento dessas ferramentas não apenas melhora a acessibilidade dos métodos, mas também acelera a transferência de tecnologia para aplicações industriais.

- O aumento do uso de métodos formais e AM em domínios críticos evidencia um campo de pesquisa promissor, especialmente em áreas relacionadas a sistemas autônomos e à segurança de infraestruturas críticas. Isso indica a necessidade de inovações que ampliem os limites tecnológicos, garantindo confiabilidade e segurança. As diversas aplicações dessas tecnologias exigem uma exploração aprofundada para assegurar sua implementação eficaz e segura em diferentes contextos.

3.4.2 Implicações para Profissionais

Para profissionais da indústria, as implicações são igualmente significativas:

- As análises destacaram a importância da adoção de procedimentos rigorosos de verificação formal para aprimorar sistemas críticos baseados em AM. A variedade de algoritmos ilustra as complexidades inerentes à aplicação do AM em ambientes críticos, ressaltando a necessidade de expertise tanto em AM quanto em metodologias de verificação. Isso enfatiza a necessidade de profissionais altamente qualificados e de colaborações eficazes entre especialistas em AM e especialistas em segurança para enfrentar os desafios de implementação e promover o desenvolvimento de avanços tecnológicos confiáveis.

- A incorporação de métodos formais no processo de desenvolvimento de sistemas críticos baseados em AM delineia um caminho para a criação de sistemas mais seguros e confiáveis. A disponibilidade de ferramentas especializadas para a verificação de

modelos, como DNNs, oferece aos profissionais os meios para avaliar e fortalecer a resiliência de suas implementações de AM. Compreender as tendências contemporâneas em verificação formal capacita os profissionais a implementar as melhores práticas no desenvolvimento e manutenção de sistemas críticos, ressaltando a importância da especialização contínua em metodologias avançadas de AM.

- A integração de métodos formais com AM em ambientes críticos ressalta a necessidade de profissionais qualificados, capazes de conceber e implementar soluções tecnológicas seguras e eficazes. O foco em sistemas autônomos e a necessidade de proteção contra falhas operacionais e riscos externos evidenciam a demanda por abordagens interdisciplinares para superar desafios complexos. Isso reforça a importância da colaboração entre especialistas em tecnologia, profissionais de cibersegurança e segurança, e especialistas em domínios específicos, todos trabalhando para garantir a resiliência e a confiabilidade operacional desses sistemas avançados.

- A identificação de áreas com pesquisa insuficiente e a avaliação do impacto de trabalhos acadêmicos podem servir como diretrizes para estudos futuros. Essa abordagem beneficia a comunidade acadêmica e oferece aos especialistas da indústria *insights* sobre desenvolvimentos emergentes e possíveis soluções para seus desafios.

3.5 Limitações e Ameaças à Validade

Diversos fatores, como o escopo do método de pesquisa, possíveis vieses na seleção dos estudos, imprecisões na extração de dados e tendências na síntese dos dados, podem influenciar os achados deste estudo. Cada um desses aspectos tem o potencial de impactar os resultados e interpretações gerais, destacando a importância de reconhecer essas limitações ao avaliar as conclusões extraídas da pesquisa.

3.5.1 Incompletude do Método de Pesquisa

Ao definir a metodologia de pesquisa, buscou-se abranger uma ampla gama de estudos, mantendo a precisão na seleção. Para isso, foi empregado o protocolo PICOC na construção da *string* de busca, o que facilitou a recuperação de inúmeros artigos de bases de dados reno-

madas. Através de um rigoroso processo de seleção em três etapas, a busca foi refinada para incluir apenas estudos relevantes sobre a interseção entre AM e métodos formais em contextos críticos, garantindo a qualidade dos dados para o mapeamento da literatura. No entanto, enfrentaram-se desafios relacionados a variações terminológicas e à ampla abrangência do tema, o que pode ter levado à omissão de alguns estudos relevantes. Isso evidencia as complexidades envolvidas na condução do mapeamento da literatura em áreas interdisciplinares e em constante evolução.

O processo de seleção de artigos está alinhado estreitamente com os objetivos estabelecidos, proporcionando uma análise aprofundada sobre o uso de métodos formais e AM em sistemas críticos. A decisão de utilizar bases de dados como ACM, IEEE Xplore, PubMed, Web of Science e Scopus foi estratégica, pois essas plataformas possuem extensos repositórios de recursos relevantes, abrangendo periódicos e anais de conferências essenciais para a compreensão da interseção dessas tecnologias em ambientes críticos. Além disso, foi utilizado um procedimento de busca suplementar por meio do *Google Scholar* para aumentar a confiabilidade dos resultados.

3.5.2 Viés no Processo de Seleção de Estudos

Foi usada uma abordagem de leitura adaptativa para mitigar viéses no processo de seleção dos estudos. Dois pesquisadores avaliaram cada artigo, e um terceiro revisor foi acionado para tomar a decisão final em casos de discordância. Além disso, foram definidos criteriosamente os critérios de inclusão e exclusão para padronizar o entendimento entre os pesquisadores, minimizando a possibilidade de interpretações divergentes. Esse método rigoroso garantiu uma seleção objetiva e representativa dos estudos, essencial para manter a integridade do mapeamento da literatura e refletir o compromisso com uma análise imparcial e abrangente do tema investigado.

3.5.3 Imprecisão na Extração de Dados

Imprecisões na extração de dados podem surgir devido à variabilidade na interpretação dos artigos pelos revisores, considerando a complexidade e diversidade dos conteúdos. Para mitigar esse risco, foram adotadas várias medidas. Inicialmente, foram definidos os itens

de dados a serem extraídos, garantindo um entendimento comum entre os pesquisadores. Em seguida, foi implementado um processo de verificação no qual pelo menos dois revisores verificaram independentemente os dados extraídos, minimizando discrepâncias. Essa abordagem colaborativa teve como objetivo reduzir imprecisões na extração e garantir a consistência e precisão das informações coletadas, fortalecendo, assim, a confiabilidade dos resultados do estudo.

3.5.4 Viés na Síntese de Dados

Na síntese de dados, o viés pode surgir da seleção subjetiva das informações durante a análise. Para mitigar esse possível viés, adotou-se uma abordagem sistemática e transparente, envolvendo pelo menos dois revisores na avaliação e interpretação dos dados. Discussões consensuais entre os revisores foram fundamentais para promover a imparcialidade e garantir que diversas perspectivas fossem consideradas. Embora tenha-se buscado minimizar o viés, é essencial reconhecer que a influência da subjetividade dos revisores na análise ainda pode ser um desafio a ser eliminado. Portanto, manter rigor e transparência no processo continua sendo fundamental para garantir a integridade e a objetividade dos resultados.

3.6 Considerações Finais

Este capítulo destaca a importância de investigar a interseção de métodos formais e AM, apresentando uma convergência significativa para melhorar a confiabilidade e segurança dos sistemas críticos. Por meio de um estudo de mapeamento sistemático da literatura, observou-se um progresso substancial na validação de vários modelos de AM, incluindo DNNs, algoritmos de aprendizado por reforço e modelos baseados em DT, integrando métodos formais.

A convergência mencionada oferece uma solução promissora para os desafios enfrentados em sistemas críticos, onde a confiabilidade e a segurança são fundamentais. O estudo identificou contribuições essenciais nesse domínio, evidenciando avanços que aprimoram a qualidade e a confiabilidade dos sistemas críticos baseados em AM.

Um dos principais focos dessa convergência é a verificação formal de DNNs, particularmente em aplicações como direção autônoma, controle de cruzeiro adaptativo e prevenção

de colisões na aviação. Métodos formais aumentam a segurança ao garantir que os modelos de AM operem conforme o esperado, minimizando o risco de falhas e comportamentos imprevisíveis. Dessa forma, eles facilitam a adoção dessas tecnologias em ambientes críticos, onde a confiabilidade é essencial.

Além disso, métodos formais desempenham um papel significativo na validação e verificação de políticas de aprendizado por reforço, garantindo propriedades críticas como a segurança do agente em cenários específicos. Isso contribui para tornar o processo de tomada de decisão dos agentes de aprendizado por reforço mais transparente e compreensível.

Também foi identificado que aprimorar a interpretabilidade dos modelos de AM, especialmente aqueles frequentemente referidos como "caixas-pretas", é essencial devido à sua falta de transparência nos processos decisórios. Essa opacidade pode representar desafios em contextos críticos, onde a confiabilidade é fundamental. Portanto, para direções futuras, a melhoria da transparência desses modelos se torna crucial, e a integração de métodos formais pode ajudar a elucidar seu funcionamento interno e reforçar a confiança em seus resultados, facilitando aplicações mais seguras em contextos de alto risco.

O presente estudo também destaca a aplicação de métodos formais na verificação de DT e RF, garantindo a correção, estabilidade e confiabilidade desses modelos. Métodos baseados em DT são mais transparentes, o que aumenta a aceitação em áreas como a saúde. A utilização de métodos formais permite que os profissionais inspirem confiança nas decisões derivadas desses modelos, promovendo um ambiente mais seguro e confiável para a implementação do AM em diversos setores.

Apesar dos avanços, desafios persistem, especialmente na melhoria das metodologias de verificação e no enfrentamento das complexidades dos ambientes dinâmicos. A ampla variedade de abordagens disponíveis destaca a dificuldade de identificar os métodos mais eficazes, enfatizando a necessidade contínua de pesquisas para abordar desafios específicos em diferentes domínios de aplicação. Além disso, a literatura precisa avançar no desenvolvimento de ferramentas que possam ser escaláveis para permitir a verificação formal de sistemas maiores e mais complexos. A verificação formal de ponta a ponta de sistemas complexos também é um avanço futuro relevante. Por exemplo, a verificação de sistemas, nos quais modelos

de AM são componentes constituintes, pode aumentar a confiança nos comportamentos que emergem de interações específicas.

Capítulo 4

Método Proposto

Este capítulo apresenta um método baseado em simulação que utiliza CPN para analisar as regras de decisão dos modelos de árvores de decisão (*Decision Tree - DT*) e florestas aleatórias (*Random Forest - RF*). O objetivo é demonstrar como essa abordagem pode melhorar a explicabilidade desses modelos. Para os modelos DT e RF, o método inclui a extração automática das regras de decisão antes de convertê-las para um modelo CPN. Em seguida, é detalhado o processo de transformação dessas regras em um modelo CPN, permitindo a simulação e análise das decisões tomadas pelos modelos de AM. A automação desse processo reduz significativamente o esforço manual e o tempo necessário para criar representações formais dos modelos, facilitando a identificação e correção de regras problemáticas.

Também é discutida a integração do Access/CPN para a simulação dos modelos gerados, destacando como essa ferramenta é utilizada para identificar, por meio de simulação, regras duplicadas e específicas, permitindo ajustes no modelo que melhoram sua explicabilidade. Além disso, uma ferramenta foi desenvolvida para facilitar o uso do método, automatizando todo o processo de geração e ajuste dos modelos CPN.

4.1 Visão geral

A Figura 4.1 fornece uma visão geral do método baseado em simulação de modelagem e análise formal, denominado RULEXTRACT/CPN. O processo envolve a implementação do

modelo de AM, a extração automática de regras de decisão, a transformação dessas regras em um modelo CPN, a análise das regras de decisão, o ajuste do modelo CPN e, finalmente, a realização de uma análise de desempenho do modelo.

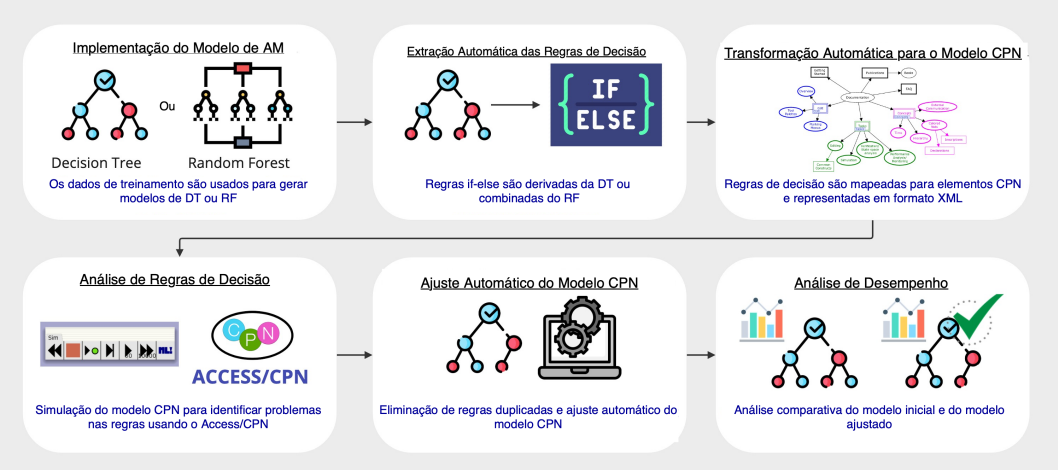

Figura 4.1: Método para modelagem formal e análise de modelos DT e RF.

4.1.1 Implementação do Modelo de AM

O processo de geração do modelo começa com o carregamento dos dados de treinamento e teste do usuário. Os usuários podem implementar um modelo de DT e RF, definindo parâmetros específicos para cada tipo de modelo. Para os modelos DT e RF, é possível ajustar configurações como a profundidade máxima da árvore ou o número de árvores na floresta.

Na implementação do método baseado em simulação, os modelos DT e RF são treinados e testados utilizando a biblioteca Scikit-learn, com o algoritmo CART sendo adotado como padrão para as DT. A métrica de acurácia é empregada para avaliar cada modelo e garantir que ele atenda ao desempenho exigido. A implementação apresentada está atualmente restrita a problemas de classificação binária, empregando a técnica de validação *hold-out*. O usuário anexa os conjuntos de treinamento e teste, podendo definir qualquer proporção desejada. Essa flexibilidade permite que os usuários ajustem a distribuição dos dados entre treinamento e teste de acordo com as necessidades específicas da aplicação de AM.

Além disso, para os modelos DT e RF o sistema oferece uma visão clara do processo de

tomada de decisão do modelo, visualizando a DT criada. Usando a biblioteca plot_tree do Scikit-learn, as DTs são convertidas em representações visuais que os usuários podem analisar para validar a lógica de decisão implementada pelo modelo. Essa visualização é útil para identificar possíveis melhorias ou ajustes necessários nos parâmetros do modelo, garantindo um processo de refinamento iterativo que leva a modelos mais robustos e precisos.

4.1.2 Extração Automática de Regras de Decisão

Uma vez treinado um modelo DT ou RF, o próximo passo é extrair as regras de decisão. Esse processo utiliza um algoritmo simples para traduzir o modelo em um conjunto de regras if-else. A Figura 4.2 representa o processo de extração automática de regras de decisão a partir de um modelo de DT ou RF. O diagrama ilustra as principais etapas desse processo:

1. **Entrada**: Um modelo DT ou RF previamente treinado, juntamente com as listas de nomes de características e classes;

2. **Processamento**: O algoritmo examina os caminhos do nó raiz até cada nó folha para formular as regras de decisão, com cada nó folha representando uma decisão específica tomada;

3. **Saída**: O algoritmo retorna um dicionário contendo as regras de decisão organizadas por classe.

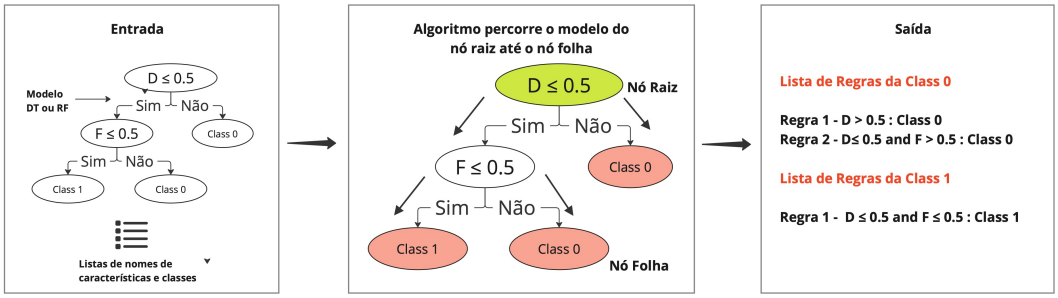

Figura 4.2: Extração Automática das Regras de Decisão

No caso de um modelo RF, o dicionário mantém as regras de decisão de cada árvore separadas, organizando-as para representar as decisões de cada árvore dentro do modelo RF. O método utiliza as regras de decisão extraídas para gerar o modelo CPN, que será empregado

para simulação do modelo. Esse processo de extração e geração de regras garante a representação precisa do conhecimento codificado no modelo de AM, facilitando a validação e a interpretação das decisões do modelo.

4.1.3 Geração Automática do Modelo CPN

Compreendendo a estrutura de lugares, transições e arcos, como apresentado no Capítulo 2, pode-se automatizar a geração do modelo CPN a partir das regras de decisão extraídas do modelo de AM (Figura 4.3). Conjuntos de treinamento e teste são as entradas para nossa geração. Assim, podemos identificar as classes-alvo e variáveis e implementar um modelo de AM. Após a extração das regras de decisão, o próximo passo é a geração automática de um modelo CPN. Um arquivo XML representa o modelo detalhando elementos e conexões. Para entender a geração do modelo CPN, é crucial compreender a representação de arcos, transições e lugares neste arquivo XML.

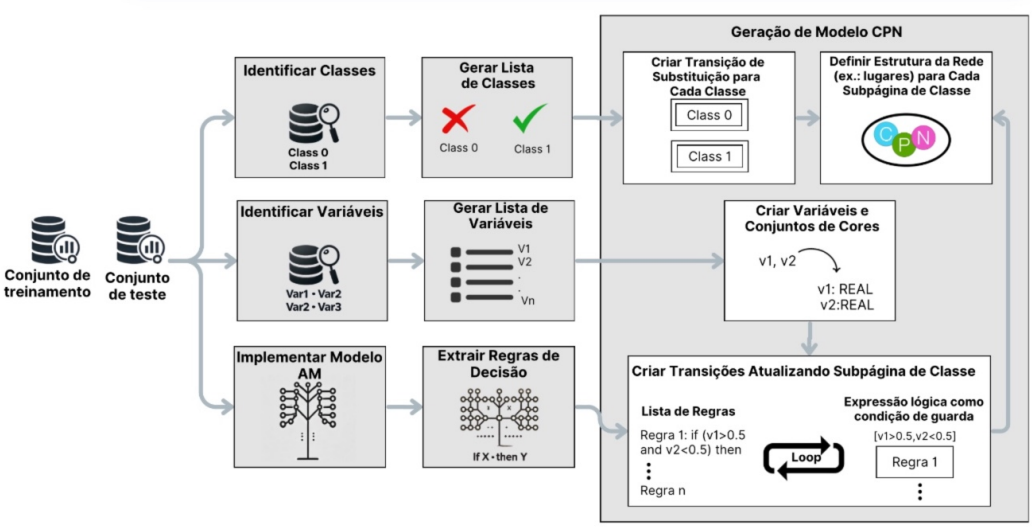

Figura 4.3: Visão Geral da Geração Automática de Modelos CPN

Os Algoritmos 2 e 3 foram definidos para gerar um modelo CPN a partir de um modelo DT e RF. Dependendo da escolha do usuário, o método baseado em simulação usará o Algoritmo 2 para DT ou o Algoritmo 3 para RF. Inicialmente, cada algoritmo recebe como entrada um dicionário contendo as regras de decisão, que foram extraídas de um modelo DT ou RF, juntamente com um conjunto de dados. Com base nessa entrada, os algoritmos seguem

etapas para incorporar as regras de decisão extraídas em um modelo CPN.

Geração a partir de DT

Primeiro, o Algoritmo 2 recupera a lista de classes do dicionário de regras de decisão e as obtém. O próximo passo é determinar o número total de variáveis no conjunto de treinamento usando uma função de forma para obter as dimensões das variáveis. Assim, o algoritmo cria uma lista de variáveis, variando de 1 até o total. Os nomes das variáveis são genéricos ($v_1, v_2, v_3, \ldots, v_n$), permitindo aplicar o modelo CPN em diversos contextos sem depender de nomes específicos de variáveis. Um loop itera pela lista de variáveis, e para cada variável, o algoritmo a insere e adiciona seu conjunto de cores correspondente no modelo CPN. Esse processo é essencial para estabelecer os elementos básicos do modelo. Em outro *loop*, o algoritmo cria a transição de substituição correspondente para cada classe na subpágina principal. O algoritmo define o lugar de cada classe na subpágina principal, seguido pelos arcos de entrada e saída.

Algorithm 2 Geração do modelo CPN mapeando regras de decisão extraídas de um modelo DT ou JRip em transições.

Entrada: Dicionário de Regras de Decisão, Conjunto de Dados **Saída:** Modelo CPN

```
1: classList ← Extrair lista de classes (dicionário de regras de decisão)
2: ruleList ← Extrair lista de regras (dicionário de regras de decisão)
3: totalVariables ← Identificar variáveis
4: variableList ← Gerar lista de variáveis
5: for variável in listaVariável do
6:     Inserir variáveis e conjuntos de cores
7: end for
8: for classe in listaClasse do
9:     Criar transição de substituição para cada classe na subpágina principal
10:    Criar lugares para cada classe na subpágina principal
11:    Criar arcos de entrada e saída na subpágina principal
12:    for regras in listaRegras do
13:        Criar transição na subpágina da classe com condição de guarda
14:        Criar lugares na subpágina da classe
15:        Criar arcos de entrada e saída na subpágina da classe
16:    end for
17: end for
18: Criar lugares na subpágina do Verificador de Rótulo de Previsão Real
19: Criar transição na subpágina do Verificador de Rótulo de Previsão Real
20: Criar arcos de entrada e saída na subpágina do Verificador de Rótulo de Previsão Real
```

Além disso, um laço interno itera por meio da lista de regras de decisão geradas pelo modelo. O algoritmo cria uma transição na subpágina da classe para cada regra, com cada regra de decisão inserida como uma condição de guarda da transição. Os lugares necessários são criados na subpágina da classe e os arcos de entrada e saída correspondentes são adicionados. Finalmente, o algoritmo cria os lugares, transições e arcos de entrada e saída na subpágina do `Verificador de Rótulo de Previsão Real`. Este processo garante que o modelo CPN esteja completo e pronto para verificar os rótulos previstos.

Na Figura 4.4 é ilustrada a página principal gerada automaticamente pelo algoritmo para o modelo DT, enquanto na Figura 4.5 é ilustrado o submódulo de comparação CPN para rótulos previstos e reais. Isso permite a validação classe por classe, analisando o total de instâncias corretamente e incorretamente classificadas para uma classe específica.

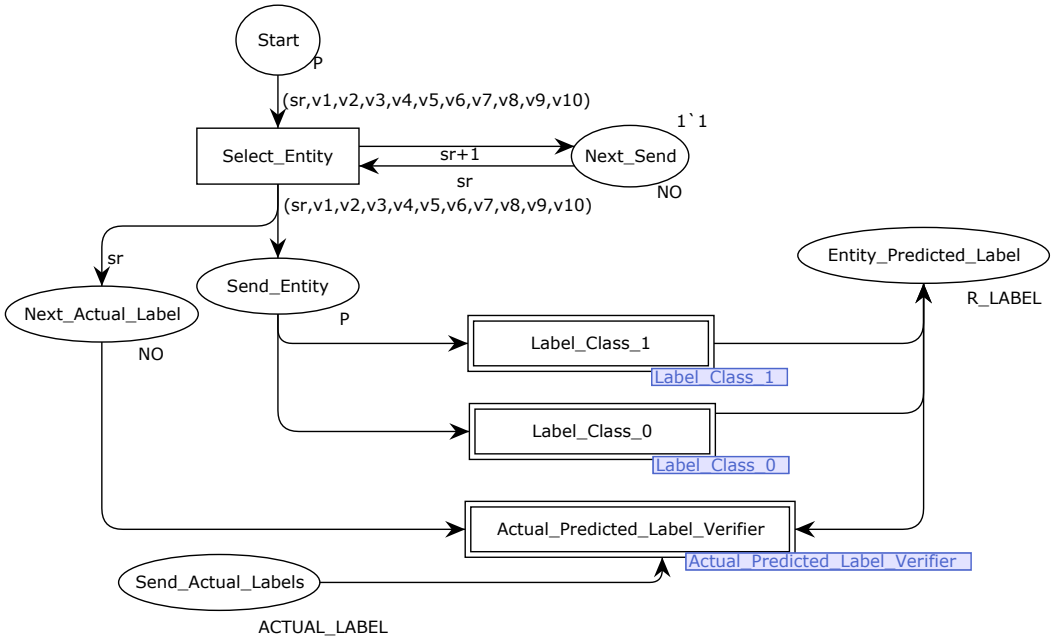

Figura 4.4: Página principal do modelo hierárquico CPN para DT.

Geração a partir de RF

Se o modelo escolhido for um RF, o método baseado em simulação executa o Algoritmo 3, recuperando a lista de árvores do dicionário de regras de decisão e as obtendo. Ele segue o mesmo procedimento das linhas 3 a 7 do algoritmo anterior. No entanto, um *loop* itera pela

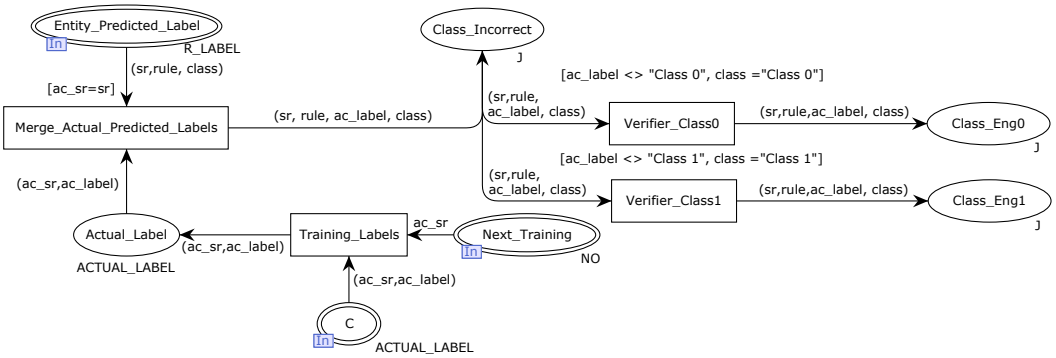

Figura 4.5: Submódulo de comparação de rótulos previstos e reais DT.

lista de árvores. Para cada árvore, o algoritmo cria a transição de substituição correspondente na página principal, os lugares para cada árvore e os arcos de entrada e saída. Outro *loop* itera pelas regras de decisão de cada árvore. O algoritmo cria uma transição na subpágina da árvore para cada regra e insere a regra de decisão como uma condição de guarda. Em seguida, os lugares necessários para ambas as classes são criados na subpágina da árvore, e os arcos de entrada e saída são estabelecidos.

Finalmente, o algoritmo cria os lugares, transições e arcos de entrada e saída na subpágina `Actual Predict Label Verifier`. Essa subpágina é crucial, pois determina a resposta final das árvores RF. Uma vez determinada a resposta final, o modelo compara os rótulos previstos e reais. Na Figura 4.6 é ilustrada a página principal gerada pelo algoritmo para o modelo RF, enquanto na Figura 4.7 é apresentado o submódulo de comparação CPN para rótulos previstos e reais.

4.1.4 Análise de Regras de Decisão

Após a geração automática do modelo CPN, o método inclui simulações usando o Access/CPN. Essas simulações produzem um relatório de execução, desempenhando um papel crucial na identificação de problemas com as regras de decisão, fornecendo uma análise detalhada de como cada regra funciona dentro do contexto do modelo. Assim, é possível observar o comportamento das regras de decisão em diferentes cenários, identificar inconsistências e redundâncias e verificar a eficácia das regras no processo de classificação.

Algorithm 3 Geração de modelo CPN por meio do mapeamento de regras de decisão extraídas de cada modelo RF em transições.

Entrada: Dicionário de regras de decisão, conjunto de dados

Saída: Modelo CPN

```
dtList ← Extrair lista de classes (dicionário de regras de decisão)
listaRegras ← Extrair lista de regras (dicionário de regras de decisão)
totalVariável ← Identificar variáveis
listaVariável ← Gerar lista de variáveis
for variável in listaVariável do
    Inserir variáveis e conjuntos de cores
end for
for DT em dtList do
    Criar a transição de substituição para cada DT na subpágina principal
    Criar os lugares para cada DT na subpágina de verificação
    Criar arcos de entrada e saída na subpágina principal
    Criar a subpágina para cada DT
    for regra em listaRegras do
        Criar a transição na subpágina DT com condição de guarda
        Criar os lugares na subpágina DT para ambas as classes
        Criar arcos de entrada e saída na subpágina DT
    end for
end for
Criar os lugares na subpágina do Verificador de Rótulo de Previsão Real
Criar a transição na subpágina do Verificador de Rótulo de Previsão Real
Criar arcos de entrada e saída na subpágina do Verificador de Rótulo de Previsão Real
```

A simulação facilita a detecção de regras duplicadas, excessivamente específicas ou incorretas, proporcionando uma visão mais clara dos pontos fortes e fracos do modelo de AM. Simular o modelo CPN pode garantir que as regras implementadas funcionem corretamente e realizem classificações precisas, permitindo ajustes necessários antes da aplicação prática do modelo.

4.1.5 Ajuste Automático do Modelo CPN

Nesta seção, será apresentado o processo de ajuste do modelo CPN, com foco na otimização das regras de decisão extraídas. No entanto, o processo de ajuste segue a mesma abordagem utilizada para os demais modelos. Na etapa de ajuste foram aplicadas apenas modificações controladas, removendo regras redundantes de forma a preservar a acurácia do modelo, garantindo um equilíbrio entre explicabilidade e desempenho.

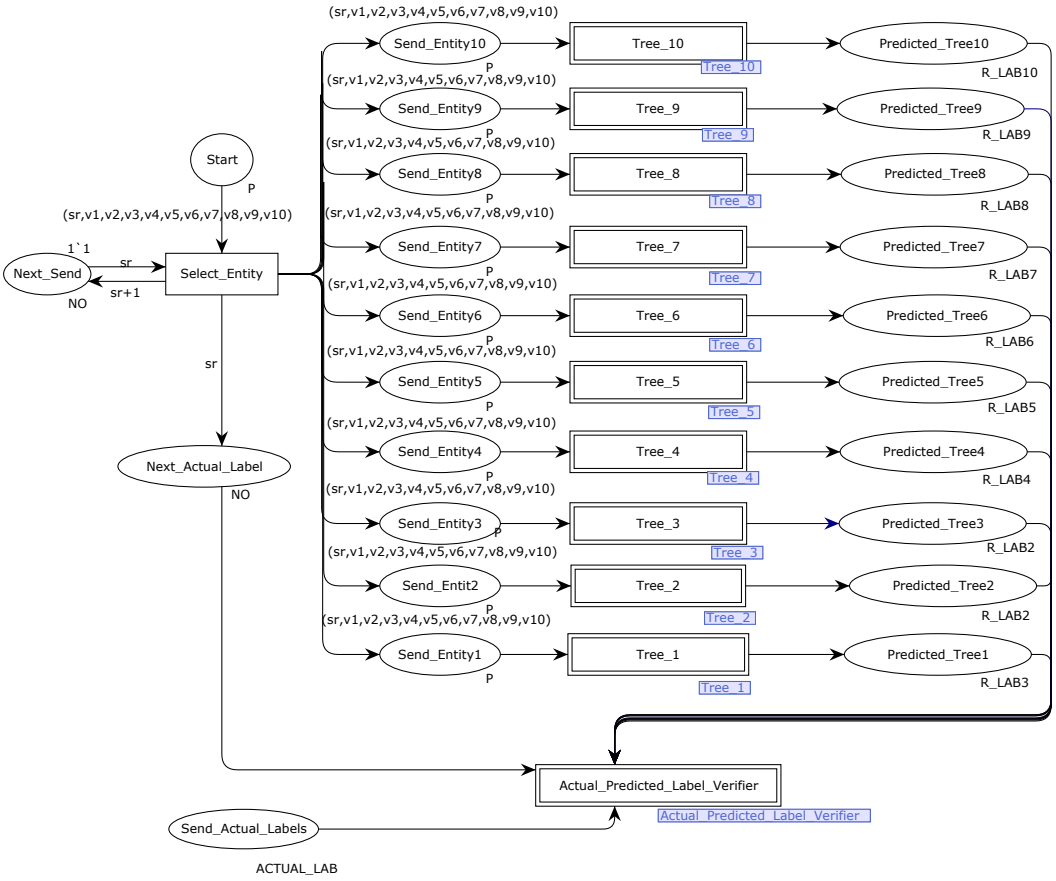

Figura 4.6: Página principal do modelo hierárquico CPN para RF.

No modelo CPN, um lugar armazena os tokens relacionados às variáveis usadas em cada regra. Cada vez que uma transição é disparada, os *tokens* são gerados e posteriormente comparados com a lista de todas as variáveis. O relatório de simulação examina esse lugar, permitindo identificar regras duplicadas.

O Algoritmo 3 detalha o processo utilizado para ajustar o modelo CPN. Ele recebe o relatório de simulação e o modelo CPN gerado automaticamente como entrada e gera como saída um modelo CPN ajustado, sem regras duplicadas.

Critérios de Ajuste Automático

O ajuste do modelo CPN segue critérios objetivos para garantir que apenas modificações aceitáveis sejam realizadas, evitando perdas na capacidade preditiva do modelo:

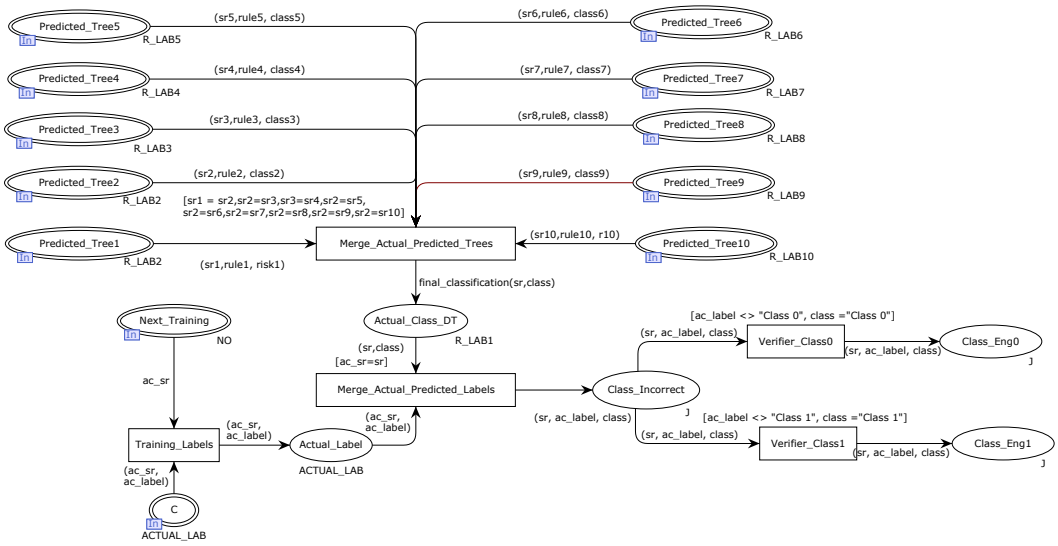

Figura 4.7: Submódulo de comparação de rótulos previstos e reais para RF.

1. Identificação de Regras Duplicadas: As regras são analisadas quanto à similaridade nas condições de decisão. Se duas regras apresentam os mesmos critérios, mas diferem apenas em pequenas variações lógicas, elas são consideradas duplicadas e consolidadas em uma única regra. Por exemplo, pode-se considerar as seguintes regras extraídas de um modelo de decisão:

   - **Regras antes do ajuste**

     – Regra 1: Se (A > 1) e (B < 0) e (C > 1) e (D > 1) então Classe 0

     – Regra 2: Se (A > 1) e (B < 0) e (C > 1) e (D $\leq$ 1) então Classe 0

   Nessas regras, os critérios de decisão são quase idênticos, diferindo apenas na condição para `D`, que assume os valores (`D < 1` ou `D` $\leq$ `1`).

   - **Processo de Consolidação**

     – Ambas as regras levam à Classe 0 e diferem apenas na condição `D`.

     – Como a condição `D` pode assumir valores menores, maiores ou iguais a 1 (`D < 1` ou `D` $\leq$ `1`) sem alterar a classe atribuída, ela se torna redundante e pode ser removida. A nova regra consolidada combina ambas, eliminando

a condição que causava duplicação.

- **Regra final após ajuste**: Se (A > 1) e (B < 0) e (C > 1) então Classe 0

Essa simplificação reduz a complexidade do modelo sem afetar sua acurácia, melhorando sua explicabilidade.

2. Priorização na Remoção de Regras: Quando são identificadas regras duplicadas, o algoritmo dá prioridade à remoção daquela que apresentou menor impacto na classificação, isto é, que cobriu um menor número de instâncias no conjunto de treinamento.

3. Identificação de Regras Específicas: Além da remoção de regras duplicadas, o método identifica regras excessivamente específicas, que cobrem um número reduzido de instâncias e podem indicar sobreajuste ao conjunto de treinamento. Essas regras são listadas no relatório final e submetidas à avaliação de um especialista para possível remoção ou ajuste.

4. Preservação da Estrutura do Modelo: O processo de ajuste mantém a estrutura lógica original do modelo CPN, garantindo que a sequência de decisão e a transição entre estados não sejam comprometidas.

Algoritmo para Ajuste do Modelo CPN

O algoritmo ajusta um modelo CPN usando um relatório de simulação como entrada, carregado a partir do arquivo `simulation_report.txt` usando a função `loadReport`. O relatório contém os dados necessários para identificar potenciais problemas nas regras de decisão. Essa função abre o arquivo de relatório, lê cada linha, extrai os dados relevantes e os armazena e retorna em uma estrutura apropriada.

Além disso, a função `loadModel` carrega o modelo CPN inicial do arquivo especificado (`initial_model.cpn`) e o retorna para uso posterior no algoritmo. Esse modelo inicial é a base para todos os ajustes necessários, garantindo que as modificações sejam aplicadas de maneira correta e consistente.

O algoritmo identifica regras duplicadas no relatório de simulação do modelo CPN usando

a função `identifyDuplicatedRules`. Essa função inicializa uma lista para armazenar as regras duplicadas e itera por todas as regras no relatório, comparando-as. Se existirem regras duplicadas, o par é adicionado a uma lista. O algoritmo então entra em um *loop* que itera pelas regras duplicadas identificadas no relatório de simulação. A função `mergeRulesInModel` atualiza o modelo CPN combinando as regras duplicadas e modificando as transições, arcos e lugares conforme necessário.

Algorithm 4 Análise do modelo CPN e remoção de regras duplicadas.

Entrada: Relatório de simulação, modelo CPN

Saída: Modelo CPN ajustado

```
simulationReport = loadReport("simulation_report.txt")
cpnModel = loadModel("initial_model.cpn")
duplicatedRules = identifyDuplicatedRules(simulationReport)
for cada par (rule1, rule2) em duplicatedRules do
    mergeRulesInModel(cpnModel, rule1, rule2)
end for
saveAdjustedModel(cpnModel, "adjusted_model.cpn")
function LOADREPORT(filePath)
    Abrir o arquivo de relatório, ler cada linha e extrair dados relevantes
    Armazenar dados em uma estrutura apropriada
    return estrutura com dados do relatório
end function
function LOADMODEL(filePath)
    Carregar o modelo CPN inicial a partir de um arquivo
    return modelo CPN
end function
function IDENTIFYDUPLICATEDRULES(report)
    Inicializar uma lista para regras duplicadas
    for cada regra1, regra2 no relatório do
        Comparar variáveis e condições das duas regras
        if variáveis e condições são idênticas then
            Adicionar (regra1, regra2) à lista de regras duplicadas
        end if
    end for
    return lista de regras duplicadas
end function
function MERGERULESINMODEL(model, rule1, rule2)
    Modificar o modelo CPN para mesclar regras duplicadas
    Atualizar transições, arcos e lugares conforme necessário
end function
function SAVEADJUSTEDMODEL(model, filePath)
    Escrever o modelo CPN ajustado em um arquivo
end function
```

Quando o algoritmo remove regras duplicadas, ele prioriza eliminar a regra específica que inicialmente classificou menos instâncias. Esse processo pode ser realizado até que não existam mais regras duplicadas. Também foram identificadas regras específicas que classificam apenas uma ou duas instâncias, mas que não são duplicadas, marcando-as no relatório para análise manual por um especialista do domínio, que decidirá se deve remover ou ajustar as regras. Portanto, o método permite a exclusão de regras específicas com consulta a um especialista. O algoritmo notifica o modelador sobre essas regras específicas, permitindo que ele busque conselhos de um especialista que possa determinar melhor se a regra deve ser removida ou ajustada.

Finalmente, o algoritmo salva o modelo CPN ajustado em um arquivo usando a função `saveAdjustedModel`, preservando todas as modificações. Essa função escreve o modelo CPN ajustado no arquivo especificado, concluindo o processo de ajuste e garantindo que todas as modificações feitas durante a execução do algoritmo sejam preservadas. Esse algoritmo fornece um método sistemático e automatizado para identificar e corrigir regras duplicadas em modelos CPN com base no relatório de simulação, garantindo a precisão do modelo resultante.

4.1.6 Análise de Desempenho

Após o processo de correção das regras, foi gerado um novo relatório de simulação para o modelo final ajustado. O método inclui a leitura do relatório de simulação e a identificação e contagem das classificações incorretas para cada classe. Contar essas ocorrências fornece uma visão detalhada de quais regras frequentemente cometem erros de classificação.

Verifica-se o número de instâncias classificadas pelas regras para identificar quaisquer regras de decisão que ainda não estejam generalizando bem, especificamente aquelas que classificam apenas alguns poucos casos. Após essa verificação, compara-se a acurácia do modelo inicial com a do modelo final ajustado usando a métrica de acurácia. Os ajustes do modelo são baseados no conjunto de treinamento, o que é essencial para garantir que o modelo aprenda e se adapte às características dos dados sem se ajustar explicitamente ao conjunto de teste. Essa abordagem garante que o modelo possa generalizar e funcionar corretamente com novos dados.

Por fim, apresenta-se os resultados comparando os modelos inicial e final. Essa comparação detalhada inclui o total de classificações incorretas para cada classe, a frequência das regras que causaram essas classificações incorretas e uma avaliação geral da acurácia do modelo após os ajustes. O método então compara a acurácia do modelo usando o conjunto de teste para garantir que não houve perda de acurácia.

4.2 Implementação

O método foi implementado como uma aplicação web utilizando Django e o Access/CPN *Simulator*. O objetivo é que os usuários insiram apenas parâmetros específicos para implementar um modelo de AM e executar simulações. Dessa forma, o processo passo a passo de geração e ajuste do modelo CPN fica oculto para os usuários finais. Os conjuntos de dados, os modelos CPN gerados automaticamente, a implementação dos algoritmos apresentados nesta tese e o código-fonte para implementar os modelos de AM foram disponibilizados no repositório Zenodo[1].

Por meio dessa implementação, os usuários podem fazer o *upload* de dados de treinamento e teste e selecionar o tipo de modelo de AM a ser analisado: DT ou RF. O Django fornece uma *interface* baseada em formulários que facilita a inserção de valores e o envio dos arquivos necessários para as simulações. Os usuários podem definir parâmetros específicos para o modelo escolhido e selecionar o número de etapas que a simulação deve executar.

Quanto mais simulações forem realizadas, maior será a precisão da análise, permitindo uma identificação mais precisa de regras duplicadas e específicas. Isso resulta em um modelo ajustado com melhor explicabilidade e generalização. A Figura 4.8 exibe a interface gráfica onde os usuários podem fazer o *upload* dos dados de treinamento e teste, selecionar o tipo de modelo e configurar os parâmetros necessários.

Uma vez definidos os parâmetros, o sistema treina e valida os modelos utilizando os dados fornecidos. A *interface* gráfica exibe os resultados desse processo (Figura 4.9). O usuário pode visualizar as regras de decisão inicialmente geradas e as melhorias aplicadas ao modelo final. Além disso, o sistema fornece métricas de avaliação detalhadas para cada modelo,

[1]https://doi.org/10.5281/zenodo.13960562

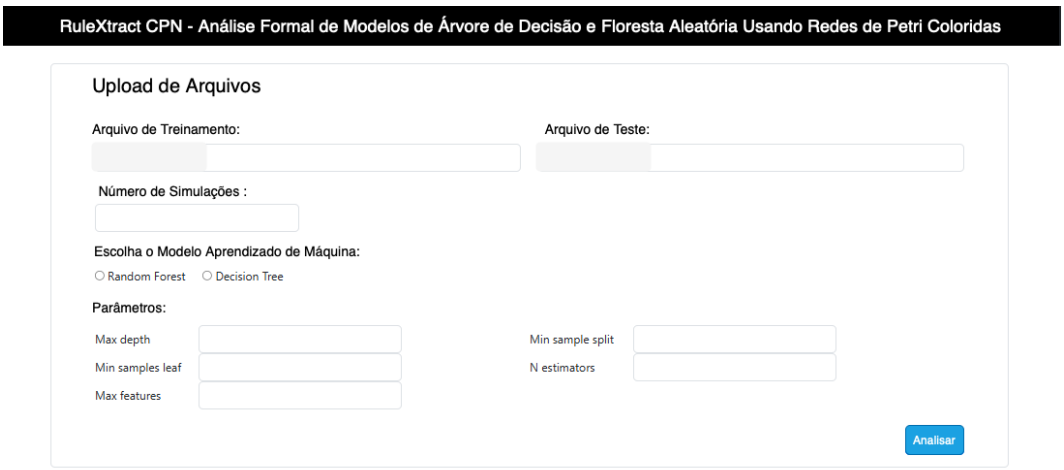

Figura 4.8: Interface web para upload de dados de treinamento e teste, seleção do tipo de modelo e configuração de parâmetros.

permitindo uma análise comparativa de desempenho antes e depois das melhorias. Uma característica diferenciada do sistema é sua capacidade de identificar e destacar as regras que levaram a classificações incorretas. Isso proporciona *insights* valiosos sobre possíveis ajustes para aprimorar o modelo. Ao final do processo, os usuários podem fazer o *download* do modelo CPN, já com todas as configurações e ajustes aplicados, possibilitando sua aplicação prática ou estudos adicionais.

4.2.1 Diagrama de Componentes

A Figura 4.10 exibe um diagrama de componentes que ilustra o fluxo de dados e a interação entre os diferentes módulos do sistema de validação formal. O diagrama representa cada componente do sistema com contêineres ou caixas, destacando as interações e chamadas de API entre eles. Inicialmente, a aplicação de página única permite que os usuários insiram valores e façam o *upload* dos arquivos de teste e treinamento necessários para as simulações, utilizando uma interface baseada em formulários que proporciona uma experiência de usuário interativa e simplificada. O componente *Django Views* processa os dados inseridos pelos usuários. Este módulo lida com os dados recebidos e executa o algoritmo de simulação selecionado pelo usuário. Após o processamento, o *Django Views*s faz uma chamada de API

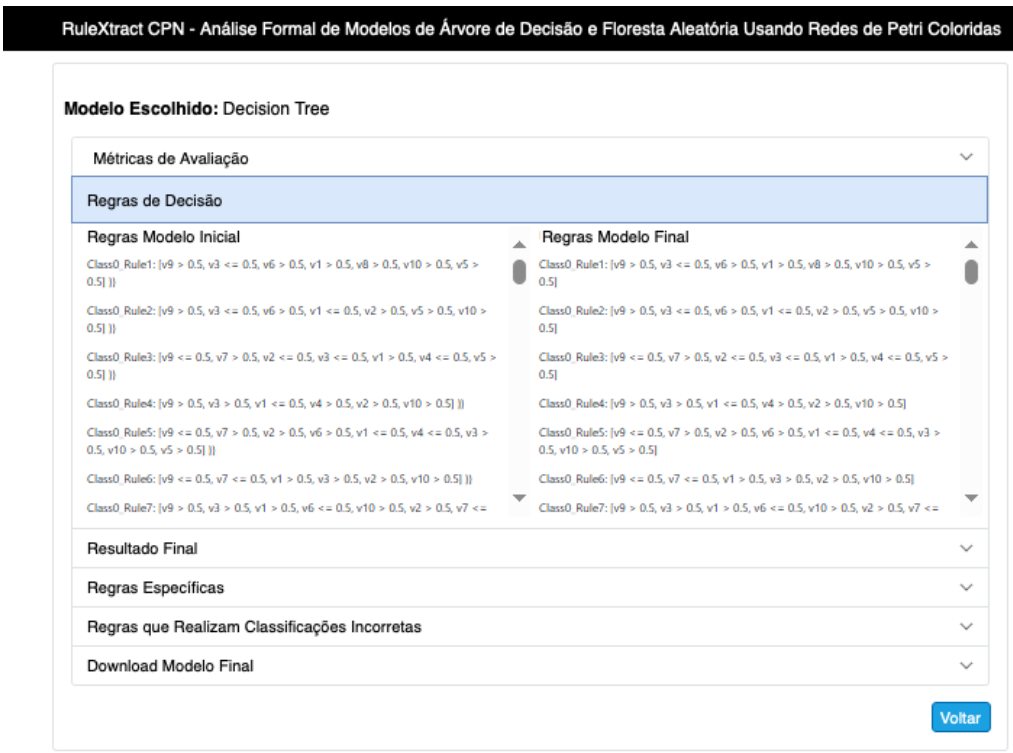

Figura 4.9: Interface exibindo regras de decisão iniciais e otimizadas com métricas de avaliação detalhadas

para o *Access/CPN Simulator*, um componente importante no sistema, enviando o modelo CPN para simulação.

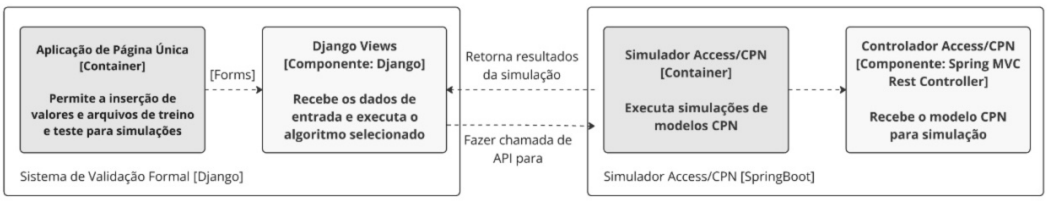

Figura 4.10: Diagrama de componentes com fluxo de dados e interação entre os diferentes módulos.

O principal papel do Simulador Access/CPN é executar simulações do modelo CPN. Ao receber o modelo CPN da API, este componente realiza as simulações necessárias para validar as regras e os processos definidos. O Controlador Access/CPN, que funciona como

um controlador REST no Spring MVC, recebe o modelo CPN e realiza a simulação dentro deste componente. Ele gerencia a lógica da simulação e garante que os resultados sejam processados e retornados corretamente para o Django Views, tornando-os disponíveis para o usuário.

Há uma separação de responsabilidades, um recurso de design essencial que aumenta a confiança na robustez do sistema. O fluxo de dados começa com a entrada do usuário, passa pelo processamento e execução do algoritmo no Django, segue para a simulação no Access/CPN, e retorna os resultados ao usuário. Uma arquitetura modular facilita a manutenção e escalabilidade do sistema, permitindo a integração de novos algoritmos ou melhorias nos componentes existentes.

4.2.2 Diagrama de Atividades

A Figura 4.11 ilustra um diagrama de atividades que descreve o procedimento para a geração automática de um modelo CPN a partir de um modelo de AM, que pode ser uma DT ou RF. O processo começa com a seleção do modelo de AM e a geração do modelo CPN correspondente. Este modelo serve como base para a simulação no Access/CPN, onde um teste rigoroso das regras de decisão do modelo é realizado. O processo de análise é de extrema importância, garantindo a precisão do modelo. Durante a simulação, o sistema verifica continuamente os resultados. Ao detectar regras duplicadas ou inconsistências, ele as remove e executa novamente a simulação.

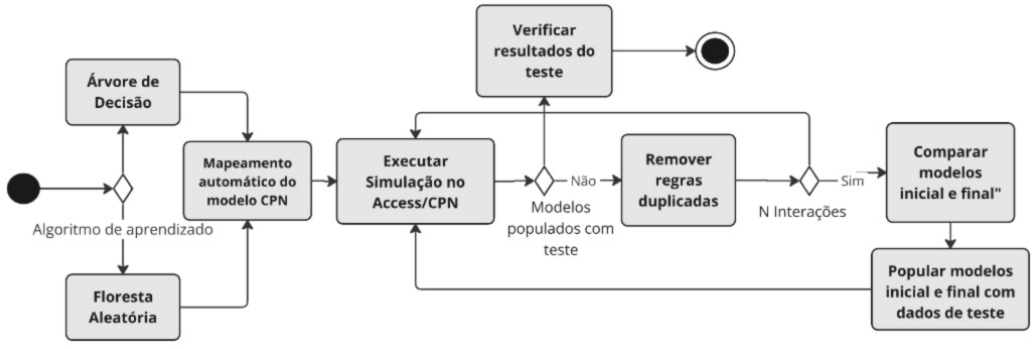

Figura 4.11: Diagrama de atividades com o procedimento para gerar automaticamente um modelo CPN a partir de um modelo de AM.

Assim, o sistema compara os modelos inicial e final para garantir que as mudanças não comprometeram a acurácia do modelo. A etapa final, que envolve o preenchimento dos modelos inicial e final com dados de teste para verificação conclusiva, é crucial para confirmar que o modelo final é aplicável em cenários do mundo real.

4.3 Considerações Finais

Neste capítulo, discutiu-se a análise e otimização de modelos de AM aplicados a sistemas críticos, com ênfase na identificação e correção de regras de decisão problemáticas. Abordou-se a geração automática de modelos CPN a partir de modelos de AM, a execução de simulações para detectar e corrigir regras duplicadas, específicas e incorretas, e a comparação de desempenho entre os modelos iniciais e ajustados.

A implementação do método como uma aplicação web facilita o uso por especialistas, permitindo a personalização dos modelos de AM e a realização de ajustes com base em dados de treinamento. A *interface* gráfica do usuário e a arquitetura modular do sistema podem proporcionar uma experiência eficiente e a possibilidade de manutenção e escalabilidade.

Os diagramas de componentes e atividades esclarecem o fluxo de dados e as interações entre os módulos do sistema, destacando a separação de responsabilidades e a comunicação eficiente entre os componentes. Esta abordagem modular facilita a integração de novos algoritmos e melhorias contínuas.

Em conclusão, este capítulo sublinhou a importância da validação e otimização de modelos de AM em sistemas críticos, ressaltando o papel essencial das simulações na correção de problemas de classificação. A abordagem apresentada assegura que os modelos finais sejam precisos, robustos e adequados para aplicações práticas, contribuindo para a confiabilidade e segurança dos sistemas críticos.

Capítulo 5

Resultados

Neste capítulo, são apresentados os resultados obtidos a partir dos experimentos realizados e discutidos sobre o método baseado em simulação. A análise incluiu dois estudos de caso para testar a acurácia do modelo, utilizando modelos de árvores de decisão (*Decision Tree - DT*) e florestas aleatórias (*Random Forest - RF*) em conjuntos de dados de COVID-19 e Influenza. Os experimentos demonstraram a capacidade do método em otimizar os modelos de AM, resultando em modelos mais simples.

5.1 Visão Geral do Cenário de Aplicação

Sistemas críticos desempenham papéis fundamentais em áreas como saúde, onde são indispensáveis para o diagnóstico de doenças, vigilância dos pacientes e gestão de dados de saúde. Com o avanço de tecnologias como o AM, esses sistemas se tornaram mais eficientes, proporcionando diagnósticos rápidos, previsões sobre a progressão das doenças e otimização de recursos médicos. Por exemplo, esses sistemas são integrados em aparelhos médicos, ferramentas de diagnóstico e sistemas de informação de saúde. A acurácia e a confiança no funcionamento desses sistemas são cruciais para fornecer cuidados adequados aos pacientes, diagnósticos precisos e estratégias de tratamento eficientes, uma vez que qualquer falha pode ter consequências graves, destacando a importância de validações abrangentes. Esses sistemas podem ser aplicados a diferentes doenças e condições médicas, como COVID-19 e

Influenza, possibilitando uma avaliação mais abrangente e precisa das condições do paciente.

A COVID-19 é uma infecção causada por um novo coronavírus, detectado pela primeira vez em dezembro de 2019 na cidade de Wuhan, na China. A transmissão ocorre principalmente por gotículas respiratórias expelidas por uma pessoa infectada, seja ao tossir ou espirrar, ou pelo contato com superfícies contaminadas. Os sintomas incluem febre, tosse e dificuldade para respirar, podendo variar de leves a graves, com a possibilidade de complicações como ventilação mecânica ou até risco de morte, especialmente em pacientes com condições pré-existentes ou idosos [41, 124]. O diagnóstico de COVID-19 envolve testes como o RT-PCR, que detectam a presença do vírus nas amostras coletadas.

Já a Influenza é uma infecção respiratória sazonal provocada pelos vírus da família *Orthomyxoviridae*, que afeta milhões de pessoas anualmente em todo o mundo. Os sintomas incluem febre alta, dores no corpo, dor de garganta, tosse seca, cansaço e dor de cabeça. A gravidade da doença varia, podendo ser leve, moderada ou grave, e pode resultar em complicações sérias, como pneumonia, especialmente em grupos vulneráveis, como crianças, idosos e pessoas com doenças preexistentes [2, 77]. A Influenza também pode ser diagnosticada por meio de testes rápidos ou RT-PCR, que identificam o vírus e permitem a diferenciação entre os tipos e cepas do vírus.

No contexto da saúde, sistemas críticos integrados com AM desempenham um papel crucial no diagnóstico de doenças como COVID-19 e Influenza. Laboratórios utilizam esses sistemas para detectar a presença de vírus em amostras biológicas, enquanto em Unidades de Terapia Intensiva (UTI), sistemas críticos monitoram pacientes com COVID-19 grave, prevendo a deterioração do paciente com base em dados de sinais vitais. Além disso, a Influenza, também se beneficia do uso de AM em sistemas de diagnóstico, permitindo análise em tempo real de dados clínicos e epidemiológicos, facilitando a previsão de surtos e aumentando a acurácia do diagnóstico. A evolução constante desses vírus exige sistemas capazes de se adaptar rapidamente, identificando novas cepas e otimizando os tratamentos, melhorando a resposta de saúde pública e reduzindo custos associados ao controle de epidemias.

5.2 Estudo de Caso

Nesta seção, são apresentados os conjuntos de dados utilizados nos dois estudos de caso, envolvendo COVID-19 e Influenza. O método baseado em simulação foi aplicado a cada conjunto de dados, envolvendo a implementação de modelos de AM, a extração e transformação das regras de decisão em um modelo CPN e a análise detalhada para identificar e eliminar regras redundantes. O objetivo é avaliar a eficácia do método na simplificação dos modelos de decisão, por meio da identificação e eliminação de regras duplicadas. A remoção dessas regras melhora a explicabilidade do modelo, tornando suas decisões mais transparentes.

5.2.1 Conjunto de Dados Covid-19

Para avaliar a eficácia do método apresentado, no primeiro estudo de caso realizou-se experimentos utilizando seis conjuntos de dados diferentes para diagnóstico de COVID-19. Cada conjunto de dados possui 11 variáveis e duas classes: 0 e 1, onde 0 indica um resultado negativo e 1 indica um resultado positivo. Para verificar e analisar as regras, a ferramenta desenvolvida para aplicar o método requer conjuntos de dados de treinamento e teste. Devido a essa necessidade, foi aplicada a validação *hold-out* utilizando a ferramenta Weka. A divisão entre os conjuntos de treinamento e teste variou conforme as características de cada conjunto de dados, garantindo um equilíbrio adequado entre os exemplos disponíveis para os dois conjuntos. Os conjuntos de dados de COVID-19 incluem:

- RT-PCR balanceado (1.832 dados): 1282 dados de treinamento e 550 dados de teste.
- RT-PCR desbalanceado (2.779 dados): 1945 dados de treinamento e 834 dados de teste.
- Teste rápido balanceado (1.290 dados): 903 dados de treinamento e 387 dados de teste.
- Teste rápido desbalanceado (7.000 dados): 4.000 dados de treinamento e 3.000 dados de teste.
- Ambos os teste balanceado (3.122 dados): 2.185 dados de treinamento e 937 dados de teste.

- Ambos os teste desbalanceado (8.000 dados): 4.000 dados de treinamento e 4.000 dados de teste.

5.2.2 Conjunto de Dados Influenza

Para o estudo de caso de Influenza, o método baseado em simulação foi avaliado usando cinco conjuntos de dados [125]. Cada um desses conjuntos de dados possui 13 variáveis. A aplicação do método seguiu a mesma abordagem utilizada para os dados de COVID-19, envolvendo a implementação de modelos de AM, a extração e transformação das regras de decisão em um modelo CPN, e a realização de uma análise detalhada das regras de decisão. Devido à necessidade de utilizar uma base de dados de treinamento e teste, foi aplicado o método *hold-out* à base de dados original, distribuindo 70% dos dados para treinamento e 30% para teste. Os dados incluem:

- RT-PCR balanceado: 1992 dados de treinamento e 832 dados de teste.
- RT-PCR desbalanceado: 1931 dados de treinamento e 972 dados de teste.
- Teste rápido balanceado: 904 dados de treinamento e 388 dados de teste.
- Teste rápido desbalanceado: 936 dados de treinamento e 401 dados de teste.
- Ambos os testes balanceados: 953 dados de treinamento e 409 dados de teste.

5.3 Validação DT

A Tabela 5.1 resume os resultados experimentais, mostrando as medidas de acurácia antes e depois dos ajustes, o número de regras removidas e o número de iterações (rodadas de simulações) para os conjuntos de dados de COVID-19 usando modelos DT. A coluna indicando o número de regras duplicadas também mostra o número de regras deletadas e ajustadas; para cada par de regras duplicadas, uma regra foi deletada e a outra foi ajustada removendo o nó de decisão que causava a duplicação.

Para visualizar melhor a redução no número de regras após os ajustes, o gráfico de barras ilustrado na Figura 5.1 compara as regras iniciais e finais para cada conjunto de dados.

Tabela 5.1: Comparação das regras e acurácia do modelo DT nos conjuntos de dados de Covid-19 antes e depois do ajuste do CPN.

Conjunto de dados	**Modelo AM**	**Acurácia do Modelo de AM**	**Acurácia Final do Modelo CPN**	**Regras Duplicaas (Regras Especificas)**	**Número de Iterações**
COVID-19 PCR desbalanceado	DT	96.16%	96.16%	-15 (8)	3
COVID-19 PCR balanceado	DT	94.72%	94.72%	-21 (8)	3
COVID-19 rápido balanceado	DT	92.50%	92.50%	-22 (8)	2
COVID-19 rápido desbalanceado	DT	97.86%	97.86%	-16 (5)	2
COVID-19 ambos os teste balanceado	DT	86.87%	86.87%	-47 (8)	4
COVID-19 ambos os teste desbalanceado	DT	93.22%	93.22%	-46 (8)	5

Observa-se um padrão consistente de otimização do modelo ao eliminar ou ajustar regras duplicadas, resultando em modelos mais explicáveis. Essa simplificação facilita a validação das regras antes de sua implementação em um sistemas de apoio à decisão (*Decision Support Systems - DSS*), garantindo maior confiabilidade e transparência no processo decisório.

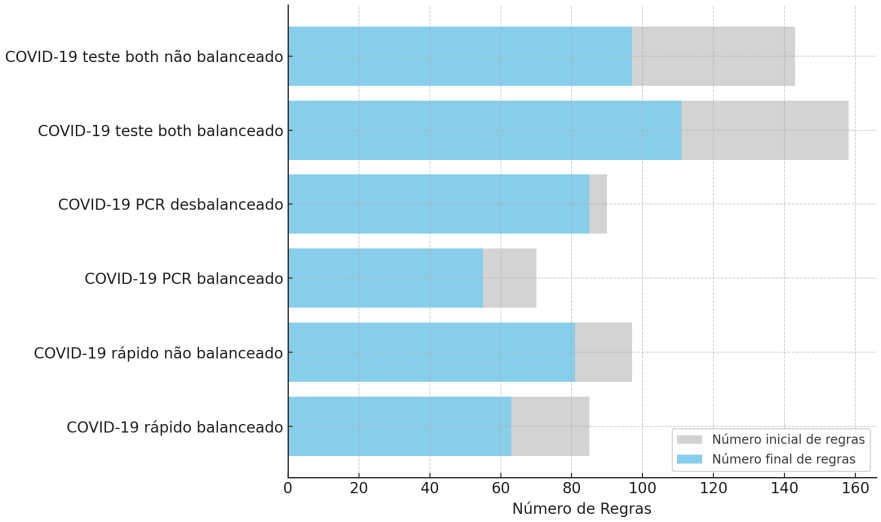

Figura 5.1: Comparação das regras iniciais e finais nos modelos DT.

No método utilizado, as regras específicas têm prioridade para remoção. A seguir, apresenta-se uma análise detalhada das reduções no número de regras para cada conjunto de dados após os ajustes.

- COVID-19 PCR desbalanceado: O modelo foi aprimorado com a redução do número de regras de 94 para 79 e a diminuição das regras específicas de 34 para 26. Isso indica que 8 das 15 regras duplicadas também eram específicas;

- COVID-19 PCR balanceado: Houve uma redução de 78 para 57 regras, onde houve a eliminação de 21 regras duplicadas após os ajustes. As regras específicas também foram reduzidas de 23 para 15;

- COVID-19 rápido balanceado: O número de regras caiu de 69 para 47, com 21 duplicadas sendo ajustadas e eliminadas. As regras específicas foram reduzidas de 15 para 7;

- COVID-19 rápido desbalanceado: O número de regras caiu de 97 para 81, com 16 regras duplicadas sendo eliminadas;

- COVID-19 ambos os testes balanceado: Houve uma redução de 158 para 111 regras, eliminando 47 regras duplicadas;

- COVID-19 ambos os testes desbalanceado: O número de regras diminuiu significativamente, de 97 para 46, com 46 regras duplicadas sendo ajustadas e removidas.

Em todos os conjuntos de dados, a acurácia dos modelos permaneceu estável após os ajustes, indicando que as modificações não afetaram a capacidade de predição dos modelos. Esses resultados demonstram que os ajustes utilizando o método foram eficazes em manter a acurácia dos modelos, ao mesmo tempo em que simplificaram as regras e aumentaram a explicabilidade dos modelos de decisão. Na Tabela 5.2 é apresentada uma amostra das regras do modelo inicial e final do conjunto de dados de COVID-19 rápido balanceado. São detalhados os ajustes realizados nas regras, incluindo um exemplo específico de duplicidade, regras ajustadas ou excluídas.

Um exemplo específico pode ser observado na regra 3, que estava duplicada com a regra 6. Ambas as regras variavam apenas pelo sinal do nó de decisão "Coriza". Nesse caso, o sistema, por meio da simulação no Access/CPN, identificou esse par de regras duplicadas e realizou o ajuste. A regra 3 foi ajustada pela remoção do nó de decisão "Coriza", que estava

Tabela 5.2: Amostra dos ajustes nas regras do modelo de COVID-19 Teste Rápido Balanceado.

Núm.	Modelo	Regra	Ação
Regra 1	Inicial	[Dispneia $\leq$ 0.5, Febre $\leq$ 0.5, Gênero $>$ 0.5, Tosse $\leq$ 0.5, Coriza $>$ 0.5, Distúrbios_do_paladar $>$ 0.5, Dor_de_garganta $>$ 0.5]	
	Final	[Dispneia $\leq$ 0.5, Febre $\leq$ 0.5, Gênero $>$ 0.5, Tosse $\leq$ 0.5, Coriza $>$ 0.5, Distúrbios_do_paladar $>$ 0.5]	Ajustada
Regra 2	Inicial	[Dispneia $>$ 0.5, Distúrbios_olfativos $\leq$ 0.5, Febre $\leq$ 0.5, Distúrbios_do_paladar $>$ 0.5, profissional_de_saúde $>$ 0.5, Dor_de_cabeça $>$ 0.5]	
	Final	[Dispneia $>$ 0.5, Distúrbios_olfativos $\leq$ 0.5, Febre $\leq$ 0.5, Distúrbios_do_paladar $>$ 0.5, profissional_de_saúde $>$ 0.5, Dor_de_cabeça $>$ 0.5]	Sem Ajuste
Regra 3	Inicial	[Dispneia $\leq$ 0.5, Febre $>$ 0.5, Distúrbios_olfativos $\leq$ 0.5, profissional_de_saúde $>$ 0.5, Coriza $>$ 0.5]	
	Final	[Dispneia $\leq$ 0.5, Febre $>$ 0.5, Distúrbios_olfativos $\leq$ 0.5, profissional_de_saúde $>$ 0.5]	Ajustada
Regra 6	Inicial	[Dispneia $\leq$ 0.5, Febre $>$ 0.5, Distúrbios_olfativos $\leq$ 0.5, profissional_de_saúde $>$ 0.5, Coriza $\leq$ 0.5]	
	Final		Excluída

causando a duplicidade e era desnecessário, enquanto a regra 6 foi eliminada. A escolha pela regra a ser ajustada ou eliminada é feita por meio da simulação no conjunto de dados de treinamento, identificando qual regra classifica menos instâncias e, portanto, deve ser eliminada. Estas reduções nas regras mostram a eficácia do método em simplificar os modelos, aumentando sua capacidade de generalização sem comprometer a acurácia, melhorando a explicabilidade dos modelos.

A Tabela 5.3 resume os resultados experimentais, mostrando as medidas de acurácia antes e depois dos ajustes, o número de regras removidas e o número de iterações (rodadas de simulações) necessários para que não houvesse mais regras duplicadas, para os conjuntos de dados de Influenza usando modelos DT.

Para visualizar melhor a redução no número de regras após os ajustes, o gráfico de barras ilustrado na Figura 5.2 está relacionado com a comparação entre as regras iniciais e finais

Tabela 5.3: Comparação das regras e acurácia do modelo DT nos conjuntos de dados de Influenza antes e depois do ajuste do CPN.

Conjunto de dados	Modelo AM	Acurácia do Modelo de AM	Acurácia Final do Modelo CPN	Regras Duplicaas (Regras Especificas)	Número de Iterações
Influenza teste rápido balanceado	DT	83.50%	83.50%	-26 (21)	2
Influenza teste rápido desbalanceado	DT	83.29%	83.29%	-25 (19)	4
Influenza ambos os testes balanceados	DT	89.73%	89.73%	-29 (19)	5
Influenza teste PCR balanceado	DT	86.76%	86.76%	-41 (22)	3
Influenza teste PCR desbalanceado	DT	87.82%	87.82%	-45 (25)	3

para cada conjunto de dados. Com base na tabela e no gráfico, é possível observar uma redução substancial no número de regras após os ajustes, indicando uma simplificação dos modelos, apresentando os seguintes resultados:

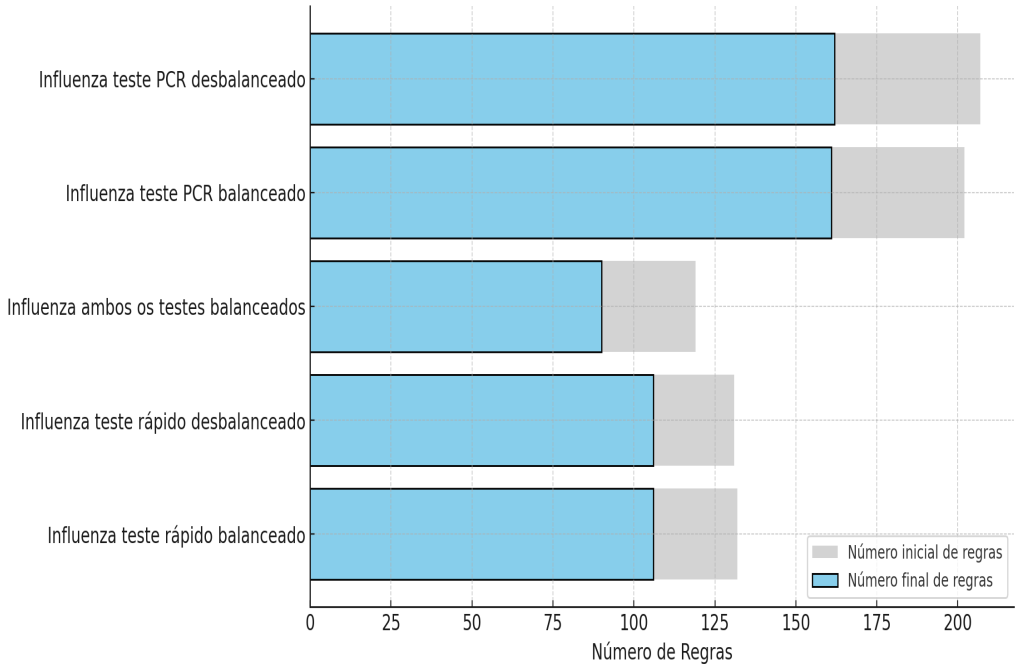

Figura 5.2: Comparação das regras iniciais e finais nos modelos DT.

- Influenza teste rápido balanceado: O número de regras foi reduzido de 132 para 106, com 26 regras duplicadas. As regras específicas diminuíram de 57 para 36.
- Influenza teste rápido desbalanceado: Houve uma redução de 131 para 106 regras,

com 25 regras duplicadas. As regras específicas caíram de 53 para 34.

- Influenza ambos os testes balanceados: O número de regras caiu de 119 para 90, com 25 regras duplicadas ajustadas e eliminadas. As regras específicas foram reduzidas de 54 para 35.

- Influenza teste PCR balanceado: A quantidade de regras diminuiu de 202 para 161, com 44 regras duplicadas. As regras específicas foram reduzidas de 54 para 32.

- Influenza teste PCR desbalanceado: O número de regras foi reduzido de 207 para 162, com 47 regras duplicadas ajustadas e eliminadas. As regras específicas diminuíram de 69 para 44.

Em todos os conjuntos de dados, a acurácia dos modelos permaneceu estável após os ajustes, indicando que eles não comprometeram a capacidade preditiva dos modelos. Esses resultados demonstram que o método baseado em simulação foi eficaz em manter a acurácia, ao simplificar as regras e aumentar a explicabilidade dos modelos de decisão. A eliminação de regras duplicadas, com destaque para a exclusão de 41 regras no teste RT-PCR balanceado e 45 no desbalanceado, juntamente com a redução de regras específicas, resultou em modelos mais simplificados.

Além de mitigar regras duplicadas e específicas, o sistema identifica regras que frequentemente geram classificações incorretas, exigindo avaliação por especialistas. Nos modelos de teste RT-PCR de Influenza, tanto balanceado quanto desbalanceado, observou-se classificações enganosas. Por exemplo, a regra 5 da classe negativa no modelo balanceado fez 14 classificações incorretas. No modelo desbalanceado, a regra 5 da classe negativa fez nove classificações incorretas, e a regra 8 da classe positiva teve 11 classificações incorretas. Esses exemplos de classificações incorretas ressaltam a necessidade do processo de revisão por especialistas, pois a acurácia é crucial para gerar confiança nos modelos de AM, especialmente em contextos de saúde, onde erros podem levar a diagnósticos incorretos e tratamentos inadequados.

Por fim, foram realizados testes com um conjunto de dados com 17.223 dados, adotando inicialmente uma divisão de 70% para treinamento e 30% para teste. No entanto, a simulação

com essa configuração não foi concluída em um período viável, tornando-se impraticável na máquina utilizada. Para contornar essa limitação, foram testadas diferentes proporções de divisão dos dados utilizando a técnica *hold-out*, conforme detalhado na Tabela 5.4.

Tabela 5.4: Impacto da divisão hold-out nos tempos de simulação.

Conjunto de Dados	Acurácia	Treinamento/Teste	Regras Iniciais	Tempo de Simulação
COVID-19 rápido não balanceado (70x30)	98,16%	12.057 / 5.166	150	Indeterminado
COVID-19 rápido não balanceado (40x60)	98,16%	6.889 / 10.335	126	Indeterminado (>16h)
COVID-19 rápido não balanceado (30x70)	98,33%	5.167 / 12.056	104	Indeterminado (>13h)
COVID-19 rápido não balanceado (25x75)	98,11%	4.306 / 12.917	92	3h 14min
COVID-19 rápido não balanceado (20x80)	97,93%	3.445 / 13.778	89	1h 45min

Os experimentos demonstraram que, à medida que o tamanho do conjunto de treinamento aumentava, o tempo necessário para concluir a simulação crescia significativamente. Para 30x70, a simulação ultrapassou 13 horas sem ser concluída, e para 40x60, não finalizou mesmo após 16 horas de execução. Com a divisão inicial de 70x30, tornou-se inviável executar a simulação na máquina utilizada. Com a configuração de *hardware* utilizada, foi possível executar a simulação apenas até a divisão 25x75, com 4.306 amostras de treinamento, levando 3 horas e 14 minutos para ser concluída.

5.4 Validação de RF

Nesta seção, serão explicados os resultados da validação utilizando o método baseado em simulação nos modelos RF. Os mesmos conjuntos de dados que foram utilizados para os modelos de DT também foram analisados com os modelos RF. O objetivo é comparar a eficácia do método na simplificação e melhoria dos modelos RF, bem como avaliar o impacto dos ajustes na acurácia e explicabilidade desses modelos.

A Tabela 5.5 compara as medidas de acurácia antes e depois dos ajustes, o número de regras removidas e o número de iterações (rodadas de simulações) para os seis conjuntos de dados de COVID-19: RT-PCR desbalanceado, RT-PCR balanceado, teste rápido balanceado, teste rápido desbalanceado, ambos os teste balanceado e ambos os teste desbalanceado.

A Figura 5.3 ilustra a redução na quantidade de regras nos modelos RF antes e depois dos ajustes. Os dados da tabela mostram que, em todos os conjuntos de dados de COVID-19,

Tabela 5.5: Comparação das regras e acurácia do modelo RF nos conjuntos de dados de COVID-19 antes e depois do ajuste do CPN.

Conjunto de dados	**Modelo AM**	**Acurácia do Modelo de AM**	**Acurácia Final do Modelo CPN**	**Regras Duplicaas (Regras Especificas)**	**Número de Iterações**
COVID-19 PCR desbalanceado	RF	95.68%	**97.12%**	-124(11)	5
COVID-19 PCR balanceado	RF	94.36%	**95.27%**	-112 (11)	5
COVID-19 rápido balanceado	RF	93.02%	**93.28%**	-147 (17)	5
COVID-19 rápido desbalanceado	RF	97.66%	97.66%	-52 (9)	2
COVID-19 ambos os teste balanceado	RF	87.83%	87.83%	-95 (12)	4
COVID-19 ambos os teste desbalanceado	RF	93.85%	94.75%	-101 (8)	2

houve uma melhoria significativa na acurácia dos modelos RF após os ajustes realizados com o método. A tabela e a figura destacam os seguintes pontos:

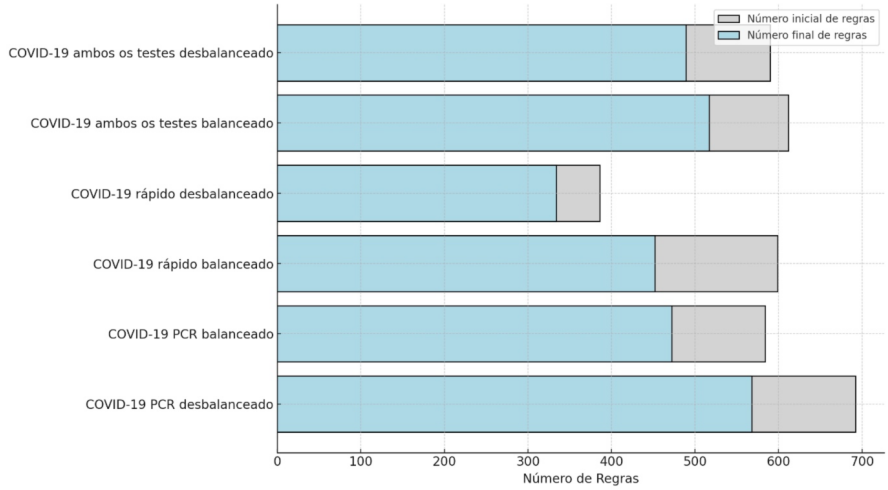

Figura 5.3: Diminuição da quantidade de regras nos modelos RF iniciais e ajustados.

- COVID-19 RT-PCR desbalanceado: O número inicial de regras foi reduzido de 692 para 568, com a remoção de 124 regras duplicadas. A acurácia inicial do modelo era de 95,68%, aumentando para 97,12% após o ajuste;
- COVID-19 PCR balanceado: O número de regras reduziu de 584 para 472 após os ajustes, identificando 112 regras duplicadas. A acurácia do modelo aumentou de

94,36% para 95,27%;

- COVID-19 rápido balanceado: A quantidade de regras foi reduzida de 599 para 452, com a remoção de 147 regras duplicadas. A acurácia subiu ligeiramente, de 93,02% para 93,28%.

- COVID-19 rápido desbalanceado: O número de regras caiu de 386 para 334. A acurácia do modelo permaneceu estável após os ajustes;

- COVID-19 ambos os testes balanceado: Houve uma redução de 612 para 517 regras, eliminando 95 regras duplicadas, sem alteração na acurácia.

- COVID-19 ambos os testes desbalanceado: O número de regras diminuiu de 590 para 489, com 101 regras duplicadas sendo ajustadas e removidas. A acurácia subiu ligeiramente, de 93,85% para 94,47%.

Os resultados mostram que a utilização do método baseado em simulação nos modelos de RF levou a uma diminuição significativa no número de regras, eliminando tanto as duplicadas quanto as altamente específicas, resultando em modelos mais simplificados. Essa simplificação é vantajosa, pois diminui a complexidade do modelo, o que pode aprimorar a interpretação e manutenção dos modelos mais acessíveis.

A Tabela 5.6 apresenta uma análise comparativa do número de regras nos modelos RF iniciais e ajustados para os 5 conjuntos de dados de Influenza: teste rápido balanceado, teste rápido desbalanceado, teste PCR balanceado, teste PCR desbalanceado e ambos os testes balanceados. Ela resume os resultados experimentais, incluindo as medidas de acurácia antes e depois dos ajustes, o número de regras removidas e o total de iterações (rodadas de simulações) realizadas nos conjuntos de dados de Influenza usando modelos RF.

- Influenza teste rápido balanceado: O número de regras foi reduzido de 882 para 688, com 194 regras duplicadas. Já as regras específicas diminuíram de 91 para 54. A acurácia inicial do modelo era de 84,02%, aumentando para 86,34% após os ajustes.

- Influenza teste rápido desbalanceado: O número de regras caiu de 876 para 687, com 189 regras duplicadas. As regras específicas foram reduzidas de 95 para 53. A acurácia

Tabela 5.6: Comparação das regras e acurácia do modelo RF nos conjuntos de dados de Influenza antes e depois do ajuste do CPN.

Conjunto de dados	Modelo AM	Acurácia do Modelo de AM	Acurácia Final do Modelo CPN	Regras Duplicaas (Regras Especificas)	Número de Iterações
Influenza rapid balanced	RF	84.02%	**86.34%**	-194 (37)	5
Influenza rapid unbalanced	RF	84.78%	**87.53%**	-189 (42)	3
Influenza both balanced	RF	90.22%	**91.19%**	-209 (32)	5
Influenza PCR balanced	RF	88.29%	88.29%	-168 (23)	4
Influenza PCR unbalanced	RF	86.11%	86.11%	-140 (25)	5

inicial era de 84,78%, subindo para 87,53% após o ajuste.

- Influenza ambos os testes balanceados: O número de regras diminuiu de 784 para 575, com 209 regras duplicadas removidas, enquanto o número de regras específicas diminuiu de 86 para 54. A acurácia passou de 90,22% para 91,19%.

- Influenza teste PCR balanceado: Inicialmente, o modelo tinha 703 regras, que foram reduzidas para 535, eliminando 168 regras duplicadas e reduzindo as regras específicas de 59 para 36. A acurácia permaneceu estável em 88,29% antes e depois dos ajustes.

- Influenza teste PCR desbalanceado: Inicialmente tinha 654 regras, que foram reduzidas para 514, eliminando 140 regras duplicadas e reduzindo as regras específicas de 69 para 44. A acurácia manteve-se estável em 86,11% antes e após os ajustes.

Na Figura 5.4 é ilustrada a diminuição da quantidade de regras nos modelos RF antes e depois dos ajustes. A análise dos dados resultou nos seguintes achados:

A aplicação do método nos modelos RF para os conjuntos de dados de Influenza resultou em uma considerável redução de regras e melhorias na acurácia. Essa simplificação facilita tanto a interpretação quanto a manutenção dos modelos, otimizando os processos de tomada de decisão. A redução da complexidade aumenta a confiança nos sistemas de diagnóstico, permitindo que as respostas em saúde sejam rápidas e precisas. Esses resultados demonstram que os ajustes realizados foram eficazes para aprimorar a qualidade dos modelos RF, contribuindo para a confiança e a eficiência dos sistemas de diagnóstico baseados em AM.

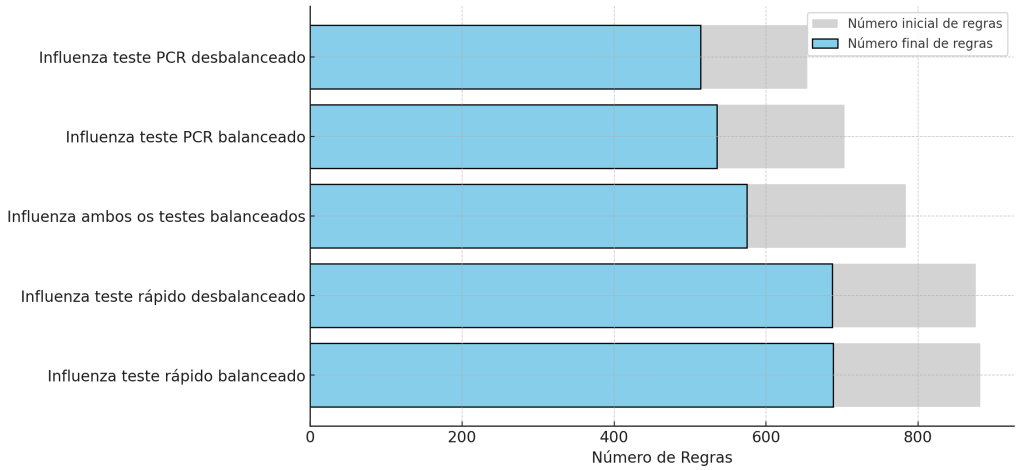

Figura 5.4: Diminuição da quantidade de regras nos modelos RF iniciais e ajustados para Influenza.

Para os conjuntos de dados Influenza RT-PCR balanceados e desbalanceados, COVID-19 rápido desbalanceado,COVID-19 ambos os testes balanceado e desbalanceado o experimento foi conduzido usando 5 DTs, enquanto 10 DTs foram usados para os outros conjuntos de dados. Essa abordagem foi necessária para reduzir a complexidade computacional e o tempo de processamento sem comprometer a qualidade dos resultados. Além disso, a variação na quantidade de árvores utilizadas (5 ou 10) também serve para avaliar o método em relação à quantidade de árvores usada, permitindo uma análise mais robusta da eficácia do método proposto na simplificação e otimização dos modelos.

5.5 Comparação entre Modelo Baseado em Regras e Baseado em DT

Os modelos de AM baseados em regras adotam uma estratégia diferente dos modelos baseados em DT na formulação de regras de classificação. Enquanto modelos baseados em DT estruturam a tomada de decisão em um formato hierárquico, onde cada decisão depende da anterior e segue um caminho específico até atingir uma classe, os modelos baseados em regras constroem um conjunto de regras independentes. Essas regras não possuem uma ordem fixa de aplicação, em vez disso, cada instância do conjunto de dados é avaliada individual-

mente em relação a todas as regras disponíveis, sendo classificada assim que atender a uma delas. Isso permite maior flexibilidade no processo de decisão, mas pode gerar modelos menos intuitivos quando comparados à estrutura sequencial e explícita das DT.

No JRip, as regras são construídas seguindo um critério específico: o algoritmo gera regras explícitas para uma das classes e, em seguida, utiliza uma única regra abrangente para classificar todas as instâncias remanescentes na outra classe. Isso resulta em uma estrutura mais enxuta para a classe principal, enquanto a classe residual pode ter uma regra final mais complexa, contendo múltiplas condições para cobrir os casos não classificados anteriormente. Nesse cenário, a etapa de extração não é necessária, pois a biblioteca Weka já retorna diretamente as regras de decisão no formato textual. Diferentemente dos modelos DT e RF, que exigem a extração das regras navegando pela estrutura das árvores, o JRip gera regras explícitas para uma das classes e utiliza uma regra abrangente para classificar as instâncias remanescentes. Esse comportamento simplifica o processo, permitindo que a conversão para um modelo CPN se concentre apenas na formatação e organização das regras. A seguir, exemplifica-se um conjunto de regras gerado pelo JRip para um problema binário:

```
1 Regra 1: v9 = 1.0 AND v3 = 0.0 AND v1 = 1.0 AND v6 = 1.0: Classe 0
2 Regra 2: v6 = 0.0 AND v2 = 1.0 AND v10 = 1.0 AND v7 = 0.0:Classe 0
3 Regra 3: v1 = 0.0 AND v5 = 1.0 AND v10 = 1.0 AND v6 = 1.0: Classe 0
4 ...
5 Regra Final: ELSE: Classe 1
```

As regras explícitas (Regra 1 a Regra 3) classificam instâncias para a Classe 0 sempre que as condições são atendidas. Já a regra final (*ELSE*) cobre todas as instâncias que não foram classificadas por nenhuma das regras anteriores, atribuindo-as automaticamente à Classe 1. Esse comportamento é diferente dos modelos DT e RF, nos quais cada classe tem suas próprias regras explícitas.

Essa abordagem reduz a redundância de regras e simplifica a estrutura do modelo, mas pode comprometer a explicabilidade, especialmente quando a regra final generaliza amplamente a segunda classe sem detalhamento específico. Esse fator é particularmente relevante em conjuntos de dados desbalanceados, onde a classe minoritária pode não ser representada de

forma tão precisa quanto na estrutura hierárquica das DT.

No exemplo a seguir, as regras representam instâncias que pertencem explicitamente à Classe 0 do conjunto de dados de Covid-19 teste rápido. Qualquer instância que satisfaça uma dessas condições será classificada como pertencente a essa classe. O JRip gera múltiplas regras explícitas para essa classe, classificando diretamente as instâncias que atendem a essas condições. Para a outra classe, em vez de um conjunto separado de regras, o algoritmo utiliza uma única regra abrangente, que cobre todas as instâncias não classificadas previamente. Embora essa abordagem reduza a redundância e torne o modelo mais compacto, ela pode dificultar a interpretação, especialmente em contextos onde a transparência do processo decisório é crucial.

```
1  Regra 1:v9 = 1.0 and v3 = 0.0 and v1 = 1.0 and v6 = 1.0
2  Regra 2:v6 = 0.0 and v2 = 1.0 and v10 = 1.0 and v3 = 1.0 and v7 =
       0.0
3  Regra 3:v1 = 0.0 and v9 = 1.0 and v4 = 1.0 and v2 = 1.0 and v5 =
       1.0 and v10 = 1.0 and v6 = 1.0
4  Regra 4:v3 = 0.0 and v9 = 1.0 and v2 = 1.0 and v4 = 0.0 and v8 =
       1.0 and v6 = 1.0 and v10 = 1.0
5  Regra 5:v3 = 0.0 and v2 = 0.0 and v1 = 1.0 and v4 = 0.0 and v9 =
       0.0 and v5 = 1.0
6  Regra 6:v1 = 0.0 and v10 = 1.0 and v3 = 1.0 and v4 = 0.0 and v9 =
       0.0 and v2 = 1.0
7  Regra 7:v7 = 0.0 and v6 = 1.0
8  Regra 8:v6 = 0.0 and v5 = 0.0
9  Regra 9:v6 = 0.0 andalso v5 = 0.0
10 Regra 10:v8 = 0.0 and v5 = 1.0 and v6 = 0.0
11 Regra 11:v8 = 0.0 and v9 = 1.0 and v2 = 1.0 and v4 = 1.0
```

Por outro lado, todas as instâncias que não foram classificadas por nenhuma dessas regras são automaticamente atribuídas à Classe 1. Essa regra pode ser representada em uma estrutura condicional "*else if*"para verificar se a instância pertence a uma das regras da Classe 0. Caso contrário, a instância será atribuída à Classe 1 por meio de uma única regra abrangente, que

pode se tornar significativamente complexa, como ilustrado a seguir.

```
1  [ not (if (v9 = 1.0 andalso v3 = 0.0 andalso v1 = 1.0 andalso v6 =
       1.0) then true
2  else if (v6 = 0.0 andalso v2 = 1.0 andalso v10 = 1.0 andalso v3 =
       1.0 andalso v7 = 0.0) then true
3  else if (v1 = 0.0 andalso v9 = 1.0 andalso v4 = 1.0 andalso v2 =
       1.0 andalso v5 = 1.0 andalso v10 = 1.0 andalso v6 = 1.0) then
       true
4  else if (v3 = 0.0 andalso v9 = 1.0 andalso v2 = 1.0 andalso v4 =
       0.0 andalso v8 = 1.0 andalso v6 = 1.0 andalso v10 = 1.0) then
       true
5  else if (v3 = 0.0 andalso v2 = 0.0 andalso v1 = 1.0 andalso v4 =
       0.0 andalso v9 = 0.0 andalso v5 = 1.0) then true
6  else if (v1 = 0.0 andalso v10 = 1.0 andalso v3 = 1.0 andalso v4 =
       0.0 andalso v9 = 0.0 andalso v2 = 1.0) then true
7  else if (v7 = 0.0 andalso v6 = 1.0) then true
8  else if (v6 = 0.0 andalso v5 = 0.0) then true
9  else if (v1 = 0.0 andalso v9 = 1.0 andalso v4 = 1.0 andalso v3 =
       0.0) then true
10 else if (v8 = 0.0 andalso v5 = 1.0 andalso v6 = 0.0) then true
11 else if (v8 = 0.0 andalso v9 = 1.0 andalso v2 = 1.0 andalso v4 =
       1.0) then true
12 else false ) ]
```

Diferentemente do JRip, os modelos baseados em DT geram regras explícitas para todas as classes, tornando a lógica de decisão mais transparente. Cada instância é classificada com base em um conjunto estruturado de condições sucessivas, permitindo a rastreabilidade do processo decisório e uma análise detalhada do caminho percorrido até a predição final. Além disso, a eliminação de regras duplicadas nos modelos DT resultou em uma estrutura mais enxuta, sem comprometer a acurácia, proporcionando um equilíbrio entre explicabilidade e desempenho. Já os modelos RF, embora inicialmente mais complexos devido ao maior número de regras geradas, demonstraram maior eficácia em cenários com distribuição de

classes desbalanceada, beneficiando-se da combinação de múltiplas árvores para melhorar a generalização.

A comparação entre os modelos JRip, DT e RF revela diferenças no desempenho e na quantidade de regras geradas. A Tabela 5.7 apresenta os resultados obtidos para diversos conjuntos de dados, destacando a acurácia e a quantidade de regras geradas pelos modelos. Os resultados evidenciam diferenças relevantes entre as abordagens, considerando tanto a complexidade do modelo quanto sua capacidade preditiva. O JRip gerou um número significativamente menor de regras e não obteve regra duplicada em nenhuma das bases de dados, resultando em modelos mais simples.

Entretanto, como mencionado anteriormente, sua estratégia de definição de regras foca em apenas uma das classes, enquanto a outra classe é inferida por uma única regra abrangente. Esse comportamento pode limitar a transparência do modelo, especialmente em conjuntos de dados desbalanceados, onde a representação da classe minoritária pode ser comprometida. Por outro lado, os modelos de DT apresentaram inicialmente um número maior de regras, mas a aplicação do método de eliminação de regras duplicadas resultou em uma redução significativa na complexidade do modelo. A redução média das regras foi de aproximadamente 25%, e essa simplificação permitiu manter a acurácia estável, garantindo um melhor equilíbrio entre explicabilidade e desempenho.

A comparação da acurácia dos modelos DT e RF antes e depois do ajuste usando o método baseado em simulação fornece insights sobre a eficácia do aprimoramento dos modelos de classificação. Como as métricas de acurácia permaneceram as mesmas para os modelos DT antes e depois dos ajustes nos modelos CPN, o método não comprometeu a qualidade. Em contraste, observou-se uma melhoria nas métricas de acurácia para os modelos RF após ajustes nos modelos CPN, especialmente nos conjuntos de dados de Influenza. Para o teste rápido balanceado de Influenza, a acurácia do modelo RF aumentou de 84,02% para 86,34%, enquanto no teste desbalanceado, subiu de 84,78% para 87,53%. A acurácia também aumentou de 90,22% para 91,19% no conjunto de dados dos testes balanceados. Para os conjuntos de dados RT-PCR balanceados e desbalanceados, a acurácia permaneceu estável, atingindo 88,29% e 86,11%, respectivamente. Essas mudanças indicam que a eliminação de regras du-

plicadas e mal generalizadas, identificadas por meio da simulação do modelo CPN, é crucial para melhorar os modelos RF.

Tabela 5.7: Comparação entre JRip, DT e RF: Acurácia e Quantidade de Regras Antes e Depois dos Ajustes

Conjunto de Dados	**Tipo de Modelo**	**Acurácia Inicial**	**Acurácia Ajustada**	**Qtd. Regras (Antes → Depois do Ajuste)**	**Nº de Iterações**
COVID-19 PCR desbalanceado	JRip	94.00%	94.00%	16 → 16	1
	DT	96.16%	96.16%	94 → 79	3
	RF	95.68%	**97.12%**	692 → 568	5
COVID-19 PCR balanceado	JRip	94.12%	94.12%	12 → 12	1
	DT	94.72%	94.72%	78 → 57	3
	RF	94.36%	**95.27%**	584 → 472	5
COVID-19 rápido balanceado	JRip	93.54%	93.54%	18 → 18	1
	DT	92.50%	92.50%	70 → 49	2
	RF	93.02%	**93.28%**	599 → 452	5
COVID-19 rápido desbalanceado	JRip	97.07%	97.07%	11 → 11	1
	DT	97.86%	97.86%	97 → 81	3
	RF	97.66%	97.66%	386 → 334	2
COVID-19 ambos os testes balanceado	JRip	87.51%	87.51%	20 → 20	1
	DT	86.87%	86.87%	158 → 111	5
	RF	87.83%	87.83%	612 → 517	2
COVID-19 ambos os testes desbalanceado	JRip	93.3%	93.3%	5 → 5	1
	DT	93.22%	93.22%	143 → 97	6
	RF	93.85%	94.45%	590 → 489	2
Influenza teste rápido balanceado	JRip	80.92%	80.92%	9 → 9	1
	DT	83.50%	83.50%	132 → 106	2
	RF	84.02%	**86.34%**	882 → 688	5
Influenza teste rápido desbalanceado	JRip	77.30%	77.30%	8 → 8	1
	DT	83.29%	83.29%	131 → 106	4
	RF	84.78%	**87.53%**	876 → 687	3
Influenza ambos os testes balanceado	JRip	91.44%	91.44%	13 → 13	1
	DT	89.73%	89.73%	119 → 90	5
	RF	90.22%	**91.19%**	784 → 575	5
Influenza PCR balanceado	JRip	85.13%	85.13%	22 → 22	1
	DT	86.76%	86.76%	202 → 161	3
	RF	88.29%	88.29%	703 → 535	4
Influenza PCR desbalanceado	JRip	79.13%	79.13%	21 → 21	1
	DT	87.82%	87.82%	207 → 162	3
	RF	86.11%	86.11%	719 → 514	5

O gráfico representado na Figura 5.5 apresenta a comparação da acurácia dos modelos JRip, DT e RF após os ajustes realizados com o método baseado em CPN. O gráfico evidencia que o RF, na maioria dos conjuntos de dados, manteve uma acurácia superior em comparação aos modelos JRip e DT. Por outro lado, o JRip apresentou uma variação maior nos diferentes conjuntos de dados, com quedas expressivas de acurácia em casos como Influenza teste rápido desbalanceado. Esse comportamento pode ser atribuído ao fato de o JRip gerar regras mais enxutas, o que o torna mais suscetível à perda de informação relevante em dados altamente desbalanceados. O DT apresentou uma acurácia semelhante à do RF na maioria dos conjuntos de dados. Embora o RF tenha mostrado superioridade em alguns cenários, o DT manteve um desempenho consistente, reforçando sua eficácia como um modelo mais explicável sem comprometer significativamente a precisão.

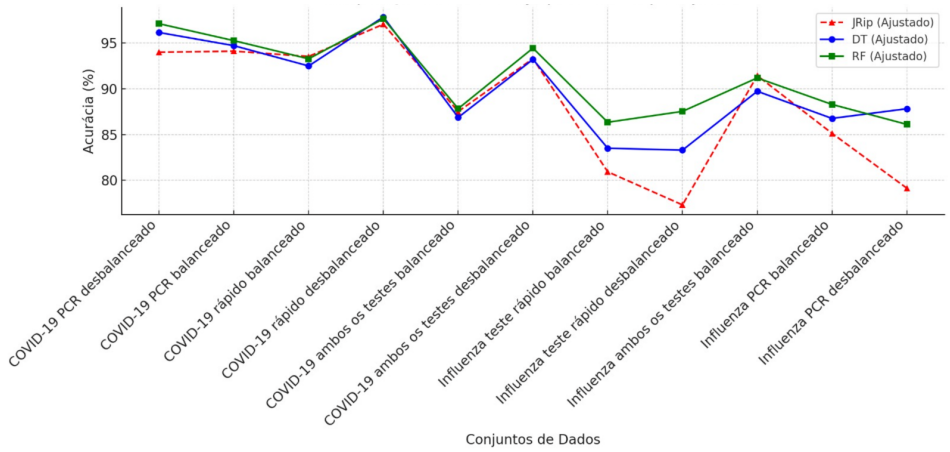

Figura 5.5: Gráfico de comparação da acurácia entre DT, RF e JRip

Essa análise fornece *insights* importantes sobre como o método pode aprimorar modelos de classificação. Por um lado, enquanto as métricas de acurácia dos modelos DT não melhoraram, houve uma redução substancial nas regras duplicadas, o que aprimorou a explicabilidade do modelo. Esse resultado valida o método, demonstrando sua eficácia em simplificar as regras de decisão sem comprometer a acurácia do modelo. Por outro lado, alguns modelos RF apresentaram maior acurácia após o ajuste.

Os resultados indicam que a escolha entre JRip, DT e RF deve considerar o contexto da aplicação. O JRip pode ser uma alternativa viável quando a prioridade é um modelo mais enxuto

e de rápida execução. No entanto, para ambientes críticos que exigem maior confiabilidade, explicabilidade e precisão, os modelos baseados em árvores, especialmente o RF, se mostram mais adequados. O DT, por sua vez, equilibra explicabilidade e desempenho, sendo uma opção relevante para cenários onde a transparência na tomada de decisão é essencial.

5.6 Considerações Finais

Neste capítulo foram abordados os resultados dos experimentos realizados com o método baseado em simulação usando CPN para melhorar a explicabilidade e a precisão dos modelos DT e RF. A análise demonstrou uma redução considerável no número de regras dos modelos DT e RF, levando a modelos mais simplificados. Foi ressaltada a importância dos sistemas críticos em setores como saúde, onde a acurácia e a confiança no funcionamento são fundamentais para assegurar segurança e eficiência. Embora os experimentos tenham sido conduzidos no contexto da saúde, o método possui potencial para aplicação em diversas outras áreas.

A aplicação do método baseado em simulação permitiu uma visualização detalhada do processo de decisão dos modelos DT, RF e JRip. A capacidade de compreender como cada decisão é tomada é essencial para garantir a confiabilidade dos modelos em contextos críticos, como diagnósticos médicos. Uma das principais contribuições do método foi a remoção eficaz de regras duplicadas, tornando os modelos mais simples e explicáveis sem comprometer sua acurácia.

Nos modelos baseados em DT, como DT e RF, a remoção de regras redundantes foi particularmente relevante. Modelos RF, por exemplo, apresentaram centenas de regras duplicadas, como observado no conjunto de dados balanceado de ambos os testes para Influenza, onde 209 regras duplicadas foram eliminadas. Essa simplificação não apenas reduz a complexidade do modelo, mas também otimiza o processo de simulação e diminui o tempo necessário para que um especialista valide as regras antes de sua incorporação a um sistema de DSS.

Além disso, o sistema também identificou regras específicas que classificam um número muito pequeno de instâncias. Nesses casos, a participação de um especialista se faz necessária para avaliar se essas regras devem ser mantidas, ajustadas ou eliminadas.

O modelo JRip apresentou um comportamento distinto, pois, diferentemente dos modelos DT e RF, ele já evita redundâncias na geração das regras, tornando desnecessária a etapa de remoção de regras duplicadas. No entanto, o método desenvolvido foi essencial para a geração automatizada do modelo CPN a partir das regras extraídas pelo JRip, permitindo uma análise estruturada e detalhada das regras. Com essa abordagem, foi possível avaliar o impacto de cada regra na classificação e verificar sua consistência dentro do modelo. Além disso, a estrutura do JRip, que gera regras explícitas para uma classe e utiliza uma única regra abrangente para a outra, pode impactar a explicabilidade em cenários desbalanceados. Esse aspecto reforça a importância de métodos formais que viabilizem a validação, organização e otimização das regras geradas, garantindo maior transparência e confiabilidade no processo de tomada de decisão.

De forma geral, a aplicação do método resultou em modelos mais simples, precisos e explicáveis. A remoção de regras redundantes, a redução da complexidade e a análise da estrutura das regras evidenciam a eficácia do método em melhorar a explicabilidade de modelos de AM. Esses avanços são fundamentais para aumentar a confiabilidade dos modelos em aplicações críticas, tornando-os mais transparentes e facilitando sua validação por especialistas antes da implementação em um DSS.

Capítulo 6

Discussões

Os resultados demonstraram a capacidade do método baseado em simulação para melhorar a explicabilidade dos modelos de árvores de decisão (*Decision Tree - DT*) e florestas aleatórias (*Random Forest - RF*). O sistema que implementa o método elimina regras duplicadas e excessivamente específicas, resultando em modelos mais simplificados. Essa simplificação, ao reduzir a complexidade, garante que os modelos sejam aplicáveis de maneira eficaz em contextos do mundo real, onde a acurácia é essencial. No entanto, ajustes adicionais podem ser realizados por especialistas para garantir que os modelos melhorem sua acurácia, especialmente em situações que exigem maior precisão nos diagnósticos. Essas melhorias podem incluir, por exemplo, a correção de constantes ou ajustes manuais em regras incorretas. A contribuição do especialista da área é crucial para garantir que as regras sejam ajustadas corretamente, aprimorando ainda mais a confiabilidade e a aplicabilidade dos modelos.

Uma contribuição significativa deste trabalho é a geração automatizada de modelos redes de Petri coloridas (Coloured Petri Nets - CPN) a partir de modelos DT e RF. Isso possibilita uma análise rápida e reduz significativamente o esforço manual e o tempo necessário para criar representações formais de modelos de AM. O uso do Access/CPN para simulações permite a identificação e remoção automáticas de regras duplicadas e específicas, melhorando a explicabilidade e a generalização dos modelos.

A comparação das métricas de acurácia antes e depois dos ajustes com CPN revelou melho-

rias nos modelos RF, como no teste rápido de Influenza balanceado, que passou de 84.02% para 86.34%, e no teste não balanceado, de 84.78% para 87.53%. No RF, a predição final é baseada no voto majoritário entre as árvores. Quando há um número par de árvores (como 10), podem ocorrer empates, resolvidos por critérios como escolha aleatória ou prioridade da classe majoritária. Os resultados mostram que a acurácia melhorou apenas nos modelos RF com 10 árvores, devido ao comportamento do modelo na resolução de empates.

Embora esta tese se concentre nos modelos de DT e RF, o método pode ser estendido para dar suporte a outros algoritmos de AM que utilizam regras de decisão, ajustando as etapas de geração automática, análise e refinamento. Além disso, embora o foco desta tese seja um cenário de saúde, o método desenvolvido pode ser aplicado a diversas áreas devido à sua abordagem independente do contexto. Inicialmente, na etapa de geração do modelo, as variáveis do conjunto de treinamento são automaticamente identificadas e convertidas em representações genéricas, como v1, v2, ..., vn. Esse processo de abstração impede que a estrutura do modelo fique vinculada a nomenclaturas específicas dos dados, garantindo maior flexibilidade e permitindo sua aplicação em diferentes domínios sem necessidade de ajustes manuais.

Após a geração, ocorre a etapa de refinamento, na qual a identificação e eliminação de regras redundantes são realizadas com base em critérios objetivos de duplicação. Esse procedimento independe da natureza dos dados ou do contexto de aplicação, pois se baseia exclusivamente na estrutura lógica das regras geradas. Dessa forma, o método pode ser facilmente adaptado para outras áreas, como finanças, segurança, indústria e transporte, onde a otimização e explicabilidade das regras de decisão são essenciais para a tomada de decisão eficiente e confiável.

No desenvolvimento desta pesquisa, optou-se por seguir o caminho da simulação em vez da verificação formal, pois, para o propósito deste estudo, a simulação se mostrou suficiente para alcançar os objetivos estabelecidos. Embora a verificação formal ofereça um nível mais profundo de análise, que é ideal para sistemas críticos, o método apresentado focou em analisar formalmente o sistema em termos de entradas e saídas, semelhante a trabalhos que utilizam verificação formal em redes neurais, onde as entradas são frequentemente abstraí-

das para subconjuntos específicos. No entanto, reconhece-se que o cenário ideal incluiria a verificação formal do modelo para garantir uma análise mais abrangente.

Portanto, como uma direção futura de pesquisa, sugere-se uma investigação mais detalhada do uso de verificação formal, onde propriedades específicas poderiam ser verificadas ao considerar um sistema crítico fim-a-fim. Nesse caso, além da simulação, a verificação usando lógica temporal e outras técnicas formais pode ser explorada para validar ainda mais a acurácia dos modelos.

As contribuições deste estudo incluem o desenvolvimento de um método para verificar regras de decisão, corrigir regras problemáticas e aplicá-lo em contextos críticos na área de saúde, além de fornecer uma análise detalhada da interseção entre métodos formais e AM. Os resultados demonstram a relevância dessa integração para aprimorar sistemas críticos, destacando sua eficácia na identificação e remoção de regras duplicadas e específicas, o que melhora a explicabilidade dos modelos. Além disso, a acurácia e a generalização podem ser aprimoradas durante ajustes manuais, permitindo uma adaptação mais refinada dos modelos às necessidades específicas de cada aplicação. Esses avanços ampliam o conhecimento na área e abrem novas direções para pesquisas futuras.

6.1 Avaliação da Explicabilidade por Profissionais da Saúde

Para avaliar a aplicabilidade e explicabilidade do método baseado em simulação em um contexto clínico, foi realizada uma colaboração com um médico. O objetivo dessa interação foi compreender como a remoção de regras duplicadas e a otimização dos modelos de decisão impactam a clareza, a confiabilidade e a validação das regras antes de sua implementação em um sistema de DSS. As perguntas formuladas para o médico foram:

1. A remoção de regras duplicadas facilitou a interpretação do modelo?
2. O processo de validação das regras antes da implementação em um DSS se torna mais rápida?

3. A simplificação das regras influencia na aceitação do modelo por médicos e outros profissionais da saúde?

A primeira questão investigou se a remoção de regras duplicadas melhorou a clareza do modelo. O médico destacou que a redundância nas regras pode dificultar a interpretação, tornando o processo de validação mais demorado e aumentando a chance de inconsistências. A eliminação de regras desnecessárias resultou em uma estrutura mais enxuta e lógica, facilitando a compreensão dos critérios de decisão. O médico observou que, antes da otimização, algumas regras incluíam variáveis que não influenciavam diretamente a decisão final, tornando o modelo mais complexo do que o necessário. Com a remoção dessas condições irrelevantes, o modelo tornou-se mais intuitivo.

A segunda questão abordou o impacto da otimização na validação do modelo antes da implementação em um DSS. O médico afirmou que, com um menor número de regras a serem verificadas, o tempo necessário para auditar o modelo é reduzido, tornando a etapa de validação mais eficiente.

A terceira questão investigou se a simplificação das regras aumentaria a aceitação do modelo por outros profissionais da saúde. O médico ressaltou que a adoção de um DSS por médicos, enfermeiros e técnicos depende da clareza e confiabilidade das recomendações geradas. Modelos excessivamente complexos tendem a ser menos utilizados, pois exigem um esforço maior de interpretação. Com a otimização, o modelo se tornou mais direto e fácil de aplicar na rotina clínica.

Além disso, o médico afirmou que acredita que a abordagem pode ser utilizada para outras patologias e domínios, desde que os dados de treinamento sejam adequados e ressaltou a importância de garantir que o sistema não direcione um diagnóstico sem considerar outras possibilidades. Como o método foca na otimização das regras de decisão antes de sua aplicação em um DSS, uma possível solução para mitigar essa preocupação seria estender o método para classificação multiclasse. Isso permitiria que as regras refinadas considerassem diferentes diagnósticos, como, por exemplo, COVID-19 e Influenza ao mesmo tempo, facilitando sua incorporação em um DSS que avalie múltiplas condições médicas antes de sugerir um resultado final. Essa melhoria abre espaço para futuras investigações, tornando o método

ainda mais útil para aplicações clínicas complexas.

6.1.1 Exemplo da Aplicação do Método na Validação Clínica

A análise inicial das regras de decisão do modelo DT, treinado com o conjunto de dados COVID-19 rápido balanceado, revelou a existência de regras redundantes que poderiam comprometer a clareza do modelo e dificultar sua validação para uso em um DSS. O método identificou e ajustou automaticamente as seguintes regras duplicadas na Classe 0 (resultado negativo):

- Regra 17 - [Dispneia ≤ 0.5, Febre ≤ 0.5, Gênero ≤ 0.5, Distúrbios Olfativos ≤ 0.5, Tosse > 0.5, Distúrbios do Paladar > 0.5]
- Regra 30 - [Dispneia ≤ 0.5, Febre ≤ 0.5, Gênero ≤ 0.5, Distúrbios Olfativos ≤ 0.5, Tosse > 0.5, Distúrbios do Paladar ≤ 0.5]

O método ajustou automaticamente as regras para reduzir redundâncias. A Regra 17 foi modificada removendo a variável redundante, enquanto a Regra 30 foi eliminada. No entanto, a simulação subsequente revelou que a Regra 17 ajustada tornou-se duplicada da Regra 6:

- Regra 17 - [Dispneia ≤ 0.5, Febre ≤ 0.5, Gênero ≤ 0.5, Distúrbios Olfativos ≤ 0.5, Tosse > 0.5]
- Regra 6 - [Dispneia ≤ 0.5, Febre ≤ 0.5, Gênero ≤ 0.5, Distúrbios Olfativos ≤ 0.5, Tosse ≤ 0.5]

Como resultado, a Regra 6 foi removida e a Regra 17 foi ajustada novamente, resultando na seguinte versão final:

- Regra 17 - [Dispneia ≤ 0.5, Febre ≤ 0.5, Gênero ≤ 0.5, Distúrbios Olfativos ≤ 0.5]

Esse processo de refinamento consolidou três regras em uma única mais concisa e informativa, facilitando sua validação antes da implementação em um DSS.

O médico avaliador destacou que a simplificação das regras reduz o esforço necessário para validá-las, pois modelos mais diretos permitem que especialistas revisem e testem sua confi-

abilidade com mais rapidez e eficiência. Ele reforçou que a ferramenta tem grande potencial para acelerar a auditoria das decisões do modelo, tornando a adoção de sistemas DSS mais segura e eficiente.

6.1.2 Perspectivas para Trabalhos Futuros

A avaliação realizada demonstrou que a otimização das regras melhora a explicabilidade dos modelos, mas para ampliar a análise do impacto do método em DSS, sugere-se que em trabalhos futuros, seja realizado entrevistas estruturadas com mais médicos, estudantes de medicina e profissionais de enfermagem.

Além disso, a dificuldade em recrutar médicos para validação reforça a necessidade de estratégias alternativas, como entrevistas com estudantes concluintes e profissionais em formação, que podem oferecer *insights* valiosos sobre a adoção do método em diferentes níveis de experiência médica.

Com base no *feedback* recebido, futuras investigações podem explorar mecanismos para incluir diagnósticos diferenciais e ajustes para reduzir vieses nos modelos de decisão, garantindo que a ferramenta seja um suporte confiável para diferentes especialidades médicas.

Assim, a colaboração com profissionais da saúde demonstrou que o método baseado em simulação facilita sua validação e interpretação, aspectos essenciais para a implementação de sistemas confiáveis de suporte à decisão clínica.

6.2 Implicações

Esta seção aborda as implicações do método baseado em simulação para diferentes públicos, incluindo pesquisadores e profissionais em indústrias críticas. O método demonstra a viabilidade e os benefícios da integração de técnicas formais com abordagens preditivas de AM, criando novas oportunidades de pesquisa e aplicações práticas em contextos críticos.

6.2.1 Implicações para Pesquisadores

Para os pesquisadores, o método abre novas oportunidades para explorar a integração de CPN com AM, como:

- melhoria da acurácia - O método baseado em simulação avança em áreas que exigem alta confiabilidade e desempenho. Ele pode servir como base para o desenvolvimento de novas técnicas e ferramentas que aprimorem a acurácia dos modelos e facilitem sua validação e implementação em sistemas críticos, como na área da saúde;

- expansão para outros algoritmos de AM - O método pode ser ampliado para abranger outros algoritmos de AM e sua aplicação em diferentes sistemas críticos;

- compreensão e ética - O aprimoramento da explicabilidade dos modelos permite que pesquisadores compreendam melhor os processos de tomada de decisão, o que é essencial para as áreas de ética e transparência em inteligência artificial. O uso de CPN na análise de modelos de AM possibilita a visualização e simulação de cada etapa da decisão, esclarecendo quais fatores influenciam os resultados. Isso facilita a verificação e validação dos modelos por especialistas e torna os sistemas mais acessíveis e compreensíveis para não especialistas; e

- promoção da pesquisa interdisciplinar - Demonstrar a aplicabilidade dos métodos formais para problemas práticos de AM ressalta a importância de novas frentes de pesquisa interdisciplinar e do desenvolvimento de metodologias que combinem rigor formal com a capacidade preditiva dos modelos.

6.2.2 Implicações para Profissionais

O uso de modelos CPN pode aprimorar a precisão de modelos de AM para apoiar profissionais que atuam em indústrias críticas, como saúde, transporte e energia. A identificação e correção de regras problemáticas antes da implementação pode ajudar a evitar erros graves e aumentar a confiança nos sistemas utilizados. Os principais impactos incluem:

- redução de erros de classificação - A melhoria na acurácia do modelo pode resultar em decisões mais confiáveis;

- aumento da confiança - Modelos mais explicáveis, com menos regras redundantes, são mais fáceis de auditar e validar. Isso aumenta a confiança dos usuários nas soluções implementadas, promovendo maior adoção e confiança nos sistemas críticos;

- desafios de tempo de processamento - Em cenários que exigem decisões rápidas, embora o tempo necessário para ajustar e validar modelos possa ser uma limitação, a capacidade computacional disponível pode garantir resultados em tempo hábil; e

- recursos computacionais - A implementação eficiente do método baseado em simulação em larga escala pode exigir investimentos substanciais em infraestrutura computacional. Planejar e justificar esses investimentos, considerando o retorno em termos de confiança e segurança do sistema, é essencial para garantir operações sustentáveis.

6.3 Limitações e Ameaças à Validade

Embora os resultados deste estudo sejam promissores, eles estão sujeitos a certas limitações e potenciais ameaças à validade. Considerando as restrições existentes, é essencial interpretar esses achados com cautela.

6.3.1 Incompletude do Método de Pesquisa

Embora o método proposto baseado em simulação mostre potencial, é importante reconhecer suas limitações em contextos práticos. Os custos de processamento e o tempo necessário para gerar relatórios de simulação, especialmente com modelos mais complexos, podem restringir a aplicabilidade do método. Esse fator deve ser uma consideração fundamental ao aplicar o método em cenários do mundo real.

Para exemplificar as exigências computacionais do método, no modelo de RF para o conjunto de dados desbalanceado de RT-PCR da COVID-19, com 1945 amostras de treinamento e 834 de teste, a simulação inicial com os dados de treinamento levou 2 horas e 19 minutos. Além disso, cinco iterações foram necessárias para identificar e eliminar todas as regras duplicadas, cada uma levando aproximadamente 1 hora e 30 minutos, totalizando um tempo de processamento de 9 horas e 49 minutos apenas para os dados de treinamento.

Esse tempo elevado ocorre devido à necessidade de traduzir as regras extraídas do modelo de AM para a estrutura de CPN e simular todas as possíveis interações entre as transições no modelo formal. Além disso, a complexidade computacional da simulação cresce de forma não linear à medida que o número de regras aumenta.

Os experimentos foram conduzidos em uma configuração de *hardware* composta por 8 GB de RAM e um processador Intel(R) Core(TM) i7-8550U. Embora essa configuração seja suficiente para muitos experimentos, ela apresenta limitações significativas para modelos mais complexos, especialmente em cenários com bases de dados maiores e maior profundidade nos modelos de decisão. Estudos recentes sugerem que a execução de simulações formais em redes neurais, por exemplo, exige *hardware* especializado, para manter tempos de resposta aceitáveis [58]. Dessa forma, futuras melhorias poderiam se concentrar na otimização da implementação do método para suportar modelos maiores e reduzir os tempos de simulação, incluindo técnicas de paralelização e uso de infraestruturas de alto desempenho.

Uma das formas de mitigar esse impacto foi restringir a análise a modelos de aprendizado supervisionado com duas classes e, no caso de RF, com no máximo 10 DT. Essas restrições foram necessárias para evitar um crescimento exponencial no número de transições a serem analisadas na simulação. Além disso, o tamanho do conjunto de treinamento foi limitado a um máximo de 4.000 amostras, garantindo que a simulação pudesse ser concluída dentro de um tempo viável no *hardware* disponível.

Embora o uso de conjuntos de dados menores possa ser questionado, estudos demonstram que eles continuam sendo relevantes na área médica, especialmente quando associados a técnicas avançadas de AM. Segundo Robinson et al. (2020) [16], mesmo com um número reduzido de amostras, é possível extrair padrões significativos e desenvolver modelos robustos, desde que haja uma análise detalhada das características dos dados e a aplicação de técnicas adequadas para evitar *overfitting*. Além disso, conjuntos de dados menores são particularmente úteis para condições raras, onde a coleta de grandes volumes de dados é inviável, mas a modelagem ainda pode fornecer *insights* valiosos para diagnóstico e prognóstico.

6.3.2 Viés na Síntese e Análise de Dados

Identificar e eliminar regras que frequentemente levam a classificações incorretas envolve decisões que podem introduzir viés. O método baseado em simulação destaca regras que fazem classificações incorretas ou que classificam poucas instâncias. No entanto, essas regras precisam ser revisadas por um especialista para determinar se ajustes adicionais são necessários.

Regras que classificam poucas instâncias foram removidas apenas em casos de duplicidade; outras regras específicas que classificam poucas instâncias permanecem inalteradas, mas são marcadas para análise posterior. Essa abordagem garante que, embora o método reduza a complexidade do modelo e aumente sua explicabilidade, decisões críticas sobre regras específicas continuem sob o julgamento de especialistas. Essa etapa é crucial para evitar o *overfitting* e garantir a acurácia do modelo em aplicações do mundo real.

6.3.3 Generalização para Outros Contextos e Aplicações

Embora o método baseado em CPN tenha se mostrado eficaz na análise e otimização de modelos baseados em DT e RF, sua aplicabilidade em outros domínios ainda precisa ser mais explorada. Atualmente, o método foi avaliado em conjuntos de dados clínicos relacionados à COVID-19 e Influenza, mas sua generalização para outros contextos, como diagnóstico de doenças crônicas ou aplicações em áreas como finanças e manufatura, requer investigação adicional.

Uma das principais questões é a adaptação do método para modelos de AM que não utilizam regras de decisão explícitas, como DNN e máquinas de vetor de suporte. Nesses casos, a extração e análise de regras se tornam mais complexas, exigindo abordagens adicionais para traduzir as decisões do modelo em um formato compatível com a estrutura de CPN.

Outro aspecto importante é a validação do método com especialistas de diferentes áreas. Durante o estudo, foi conduzida uma avaliação com um médico para validar a aplicabilidade no contexto clínico. Porém, para garantir uma implementação mais abrangente, será necessário expandir essas avaliações para profissionais de outras áreas. Estudos futuros devem explorar como a interpretação de regras otimizadas por CPN pode ser compreendida por especialistas

de setores distintos, garantindo que a técnica seja ajustável a diferentes aplicações.

6.3.4 Generalização do Modelo e *Overfitting*

Os ajustes realizados nos modelos CPN visam aprimorar a explicabilidade, mas o risco de *overfitting* aos dados de treinamento persiste, principalmente devido a regras excessivamente específicas que classificam apenas uma ou duas instâncias. O método prioriza a eliminação dessas regras quando são duplicadas e sinaliza quando elas afetam apenas uma quantidade muito restrita de instâncias. No entanto, as regras problemáticas devem ser revisadas manualmente por especialistas para equilibrar a simplificação automática com a acurácia do modelo. Esse ajuste manual deve ser feito com cautela, para evitar o risco de *overfitting*, garantindo que a generalização do modelo seja preservada enquanto se mantém sua precisão.

6.3.5 Explicabilidade e Complexidade

A redução de regras duplicadas e específicas melhora a explicabilidade do modelo e reduz sua complexidade. O método baseado em simulação elimina automaticamente apenas as regras duplicadas, o que não impacta o desempenho, pois essas redundâncias não contribuem para a capacidade preditiva do modelo. Por outro lado, as regras específicas — aquelas que classificam apenas uma ou duas instâncias — podem capturar nuances importantes dos dados e, por isso, sua remoção deve ser realizada manualmente por especialistas. Essa revisão manual é crucial para manter o equilíbrio entre a simplificação do modelo e sua acurácia, garantindo que o modelo final seja abrangente e adequado para capturar as sutilezas dos dados sem comprometer sua capacidade de generalização.

6.4 Escopos de Uso do Mundo Real

A integração de métodos formais e técnicas de AM desempenha um papel crucial em sistemas críticos em diversas indústrias e contextos. A saída do método baseado em simulação, um conjunto de regras de decisão, é essencial em ambientes onde acurácia e confiança no funcionamento são essenciais.

No setor de saúde, as regras de decisão derivadas de modelos de AM podem ser integradas

em hospitais e clínicas para automatizar diagnósticos e sugerir tratamentos com base em dados de pacientes, históricos médicos e resultados de exames. A utilização dessas regras em sistemas de DSS pode auxiliar médicos a identificar e tratar doenças como problemas cardíacos e diabetes com maior rapidez e acurácia, diminuindo a possibilidade de erros humanos e agilizando o processo de tratamento.

No âmbito financeiro, as regras de decisão podem ser aplicadas para a detecção de fraudes e na automação de decisões de crédito. Algoritmos de AM que identificam padrões suspeitos em transações ajudam a prevenir fraudes financeiras, enquanto regras aplicadas à avaliação de crédito agilizam os processos de empréstimo, tornando-os mais rápidos e justos, minimizando preconceitos humanos.

Essas aplicações sublinham a importância de ter regras de decisão rigorosamente testadas, particularmente em contextos onde as falhas podem resultar em consequências graves. A avaliação dessas regras por meio de simulação garantem que elas operem corretamente em todas as situações previstas, proporcionando uma camada adicional de segurança e confiança no funcionamento aos sistemas críticos.

6.5 Considerações Finais

Neste capítulo foi discutido o método baseado em simulação utilizando CPN para melhorar a explicabilidade e a acurácia dos modelos DT e RF. O objetivo foi demonstrar como o método pode ser aplicado para melhorar a explicabilidade, acurácia e generalização de modelos de DT e RF. Os experimentos com conjuntos de dados de COVID-19 e Influenza mostraram uma redução significativa no número de regras e uma melhoria nas métricas de acurácia para os modelos RF. A discussão destacou as implicações do método para pesquisadores e profissionais, enfatizando a relevância da integração de técnicas formais com AM. Também foram abordadas as limitações do método e as ameaças à validade, sugerindo a necessidade de revisão das regras específicas por especialistas.

Como direção futura de pesquisa, sugere-se uma investigação mais detalhada do uso de verificação formal para sistemas críticos, visando melhorar ainda mais a robustez e a acurácia dos modelos. Este capítulo, portanto, estabelece uma base sólida para a aplicação prática e a

pesquisa contínua na modelagem formal de modelos de AM.

Capítulo 7

Conclusões e Futuras Direções de Pesquisa

Uma das principais vantagens dos modelos baseados em árvores de decisão (*Decision Tree - DT*) é a possibilidade de interpretar diretamente as regras de decisão. No entanto, à medida que o número de regras cresce, o modelo pode se tornar excessivamente complexo, reduzindo sua explicabilidade. Essa complexidade é agravada pela presença de regras duplicadas, que não apenas aumentam o tempo de processamento, mas também dificultam a compreensão do modelo.

Nesse contexto, a melhoria das regras de decisão desempenha um papel crucial na qualidade dos modelos empregados em sistemas de apoio à decisão (*Decision Support Systems - DSS*). Para que esses sistemas sejam confiáveis, especialmente em aplicações críticas como a saúde, é essencial garantir um equilíbrio entre acurácia e explicabilidade. O número de classificações de falsos positivos e a transparência das decisões são aspectos fundamentais que impactam diretamente a adoção de DSS clínicos.

Além disso, para que as regras de decisão possam ser incorporadas a um DSS, elas precisam passar por uma análise detalhada por especialistas, garantindo que estejam alinhadas com o contexto clínico e a prática médica. Quando há um grande número de regras duplicadas, essa revisão se torna mais complexa e demorada, aumentando a dificuldade de validação. Por-

tanto, a eliminação de regras redundantes não apenas melhora a explicabilidade dos modelos, mas também facilita sua integração em DSS clínicos.

Diante desse desafio, esta tese apresentou o método RuleXtract/CPN, baseado em CPN, para aprimorar a explicabilidade e a eficiência de modelos de AM fundamentados em DT e florestas aleatórias (*Random Forest - RF*). O método demonstrou sua eficácia na eliminação de regras duplicadas, na identificação de regras específicas e regras incorretas que poderiam comprometer a generalização dos modelos. Com isso, viabiliza-se a construção de modelos mais enxutos e organizados, facilitando sua análise e revisão.

Um aspecto significativo do método é a geração automatizada de modelos CPN a partir de modelos DT e RF, que se mostrou crucial para a eficiência do processo, reduzindo o tempo e o esforço manual. Como resultado, o método facilita a compreensão dos modelos aumentando a confiança para aplicações no mundo real.

As descobertas deste estudo não apenas avançam o estado da arte, mas também abrem novas direções para pesquisas futuras, enfatizando a importância da combinação entre métodos formais e AM. O uso da simulação via CPN proporcionou uma maneira estruturada e rigorosa de identificar padrões problemáticos nos modelos, permitindo uma análise mais aprofundada por meio de simulações passo a passo, possibilitando uma compreensão detalhada do comportamento do modelo em diferentes cenários.

Nos experimentos realizados, a remoção de regras duplicadas resultou em modelos mais simples e organizados, sem perda de acurácia. Além disso, a redução do número de regras impacta positivamente a viabilidade da revisão humana, facilitando a validação antes do uso em DSS. Esse aspecto é essencial para garantir que sistemas baseados em AM sejam não apenas precisos, mas também confiáveis e aplicáveis na prática médica.

A tese contribui para a interseção entre métodos formais e AM, explorando como técnicas de modelagem formal podem ser empregadas para aumentar a confiabilidade e a transparência de sistemas críticos. O uso de CPN possibilitou a verificação automatizada da consistência dos modelos, algo que tradicionalmente exige inspeções manuais demoradas. As descobertas deste estudo não apenas avançam o estado da arte, mas também abrem novas direções para

pesquisas futuras, enfatizando a importância da combinação entre métodos formais e AM.

Além de validar e expandir os resultados de trabalhos anteriores que destacam a importância da simplificação dos modelos para aumentar sua explicabilidade, os achados desta tese reforçam que essa simplificação não compromete a acurácia. Estudos prévios já haviam indicado que DT mais enxutas oferecem explicações mais acessíveis sem perda significativa de desempenho preditivo, o que foi validado pelos experimentos realizados.

Os resultados obtidos também têm implicações diretas na adoção de DSS clínicos. A validação das regras ajustadas por profissionais da saúde é essencial para a implementação segura de tais sistemas. Durante as interações com especialistas, foi observado que modelos com menor redundância são mais fáceis de interpretar e revisar, promovendo maior aceitação da tecnologia no ambiente clínico. A confiabilidade dos sistemas de AM em saúde está diretamente relacionada à sua capacidade de fornecer recomendações transparentes e justificáveis, e a eliminação de regras problemáticas contribui para esse objetivo.

Embora o método baseado em simulação apresente diversas vantagens, é necessário reconhecer suas limitações e apontar caminhos para aprimoramentos futuros. Algumas direções para pesquisa incluem:

- Adaptação a outros tipos de algoritmos de AM: Atualmente, o método se aplica a DT e RF. No entanto, sua extensão para modelos baseado em regras, máquinas de vetor de suporte, redes neurais e aprendizado por reforço pode ampliar seu impacto.
- Aprimoramento da infraestrutura computacional: A simulação de modelos maiores pode demandar melhorias na implementação do método para garantir escalabilidade.
- Otimização do método RF para lidar com um maior número de DT.

Os resultados deste estudo oferecem novas perspectivas para pesquisas futuras, demonstrando que a combinação de métodos formais com AM tem potencial para transformar a forma como sistemas críticos são validados e implementados. Além de aprimorar a explicabilidade dos modelos, essa abordagem pode servir como base para o desenvolvimento de ferramentas mais confiáveis para DSS em diversos setores, especialmente na saúde.

7.1 Direções para Pesquisas Futuras

Como mencionado anteriormente, uma oportunidade para trabalhos futuros é expandir o método para abranger outras técnicas de AM além de DT e RF, como máquinas de vetor de suporte e redes neurais. Atualmente, o método baseado em simulação requer um conjunto de regras de decisão como entrada para a geração automática de modelos CPN. No entanto, é possível explorar soluções para a extração de regras de outros modelos de AM. Por exemplo, Han *et al.* [42] propôs uma solução para a extração de regras de modelos máquinas de vetor de suporte com base em uma abordagem de aprendizado em conjunto. Em contraste, Chakraborty *et al.* [14] apresentou um algoritmo para a extração de regras de redes neurais. Portanto, pretende-se investigar o uso de tais soluções em trabalhos futuros para aprimorar o suporte do método para outros modelos de AM além das DTs. Além disso, o método pode ser utilizado para melhorar os resultados de classificadores baseados em regras implementados usando algoritmos como PART e PRISM [133].

Atualmente, o método permite que os usuários carreguem conjuntos de dados de treinamento e teste. Para trabalhos futuros, planeja-se aprimorar o sistema para que os usuários possam enviar apenas um conjunto de dados, com o sistema realizando a validação *hold-out* e permitindo a especificação da proporção de divisão dos dados como um parâmetro. Além disso, planeja-se também incluir a opção para os usuários anexarem diretamente um modelo DT ou RF pré-treinado, em vez de carregar os conjuntos de dados. Assim, o usuário pode escolher na interface gráfica do usuário entre carregar diretamente um modelo DT e RF, enviar um único conjunto de dados para o sistema realizar a validação *hold-out* ou carregar os conjuntos de dados de treinamento e teste.

Para fortalecer a explicabilidade do método e complementar as análises quantitativas, futuros estudos devem incorporar avaliações qualitativas detalhadas. Entrevistas com médicos, estudantes de medicina e profissionais de enfermagem podem fornecer *insights* sobre a explicabilidade das regras ajustadas e sua aplicabilidade clínica. A análise dessas entrevistas ajudará a identificar desafios na adoção de modelos explicáveis e aprimorar a comunicação dos resultados.

Além disso, estudos de caso poderão demonstrar como a eliminação de regras redundantes

impacta a clareza e a eficácia dos modelos em diferentes cenários clínicos. Experimentos com usuários finais, incluindo questionários qualitativos, permitirão avaliar a compreensão das regras otimizadas, garantindo que o modelo final seja não apenas preciso, mas também explicável e útil para a prática médica.

7.2 Publicações

Como resultado inicial desta pesquisa, foram publicados dois trabalhos que incentivaram o uso de CPN para aumentar a confiança em sistemas baseados em AM. O primeiro estudo utilizou uma base de dados limitada de Doença Renal Crônica (DRC), abordando problemas relacionados a conjuntos de dados desbalanceados e limitados. Comparou métodos de validação e técnicas de aumento de dados, revelando que, mesmo em bases de dados restritas e com DT pequenas, podem existir regras de decisão duplicadas [108]. O segundo estudo utilizou uma base de dados extensa sobre COVID-19 para priorizar pacientes sintomáticos para testes. Este estudo destacou a necessidade de automatizar a verificação e o ajuste das regras para modelos mais complexos [124].

1. Silveira, A. C. D., Sobrinho, Á., Silva, L. D. D., Costa, E. D. B., Pinheiro, M. E., & Perkusich, A. (2022). Exploring early prediction of chronic kidney disease using machine learning algorithms for small and imbalanced datasets. Applied Sciences, 12(7), 3673 (*published*) [108].

2. Viana dos Santos Santana, Í., CM da Silveira, A., Sobrinho, A., Chaves e Silva, L., Dias da Silva, L., Santos, D. F., & Perkusich, A. (2021). Classification models for COVID-19 test prioritization in Brazil: Machine learning approach. Journal of medical Internet research, 23(4), e27293 (*published*) [124].

Além disso, após o desenvolvimento da pesquisa e do método, foram submetidos dois artigos: descrevendo o método baseado em simulação e os resultados dos experimentos realizados. O primeiro artigo foi publicado e o segundo foi aceito para publicação.

1. Silveira, A. C. M., Sobrinho, Á., Dias da Silva, L., Santos, D. F. S., Nauman, M., & Perkusich, A. (2025). Harnessing coloured Petri nets to enhance machine learning: A

simulation-based method for healthcare and beyond. Simulation Modelling Practice and Theory. https://doi.org/10.1016/j.simpat.2025.103080 (*published*) [23].

2. Silveira, A. C. M., Sobrinho, Á., Dias da Silva, L., Santos, D. F. S., Nauman, M., Perkusich, A. Formal Analysis of Tree-Based Machine Learning Models Using Coloured Petri Nets. LatinIoT-2025 (*to be published*).

Como parte dos avanços da pesquisa, também foi realizado o registro do *software* que implementa o método proposto, garantindo a proteção intelectual e facilitando a disseminação e uso da ferramenta desenvolvida:

1. RuleXtract/CPN: Um Software para a Análise Formal de Sistemas Críticos Baseados em Árvore de Decisão e Floresta Aleatória (2024). Programa de Computador.
    - Autores: Andressa C. M. da Silveira, Álvaro Sobrinho, Leandro Dias da Silva, Angelo Perkusich.
    - Número do registro: BR512024004513-9.
    - Instituição de registro: INPI - Instituto Nacional da Propriedade Industrial
2. Uma Ferramenta para Conversão de Modelos Baseados em Árvore de Decisão para Modelos de Redes de Petri Coloridas (2024). Programa de Computador.
    - Autores: Andressa C. M. da Silveira, Álvaro Sobrinho, Leandro Dias da Silva, Angelo Perkusich.
    - Número do registro: BR512024004513-9.
    - Instituição de registro: INPI - Instituto Nacional da Propriedade Industrial

7.3 Considerações Finais

A combinação de métodos formais e AM demonstrou ser uma abordagem promissora para garantir modelos mais explicáveis e confiáveis. O método desenvolvido nesta tese avança o estado da arte ao permitir uma análise automatizada e sistemática de regras de decisão,

promovendo maior transparência em modelos de AM aplicados a sistemas críticos.

Os resultados obtidos não apenas validam a eficácia da abordagem desenvolvida, mas também destacam novas direções para pesquisas futuras, reforçando a importância de soluções que combinem verificação formal e AM para garantir a segurança e confiabilidade de sistemas inteligentes.

Bibliografia

[1] Ahsan Abbas, Muhammad Arslan, Govinda Kumar, and Kashif Saghar. Formal verification and development of living assistance system. In *2021 International Bhurban Conference on Applied Sciences and Technologies (IBCAST)*, pages 420–425, 2021.

[2] Charles Oluwaseun Adetunji, Olulope Olufemi Ajayi, Muhammad Akram, Olugbemi Tope Olaniyan, Muhammad Amjad Chishti, Abel Inobeme, Seyi Olaniyan, Juliana Bunmi Adetunji, Mathew Olaniyan, and Samson Oluwasegun Awotunde. Medicinal plants used in the treatment of influenza a virus infections. *Medicinal Plants for Lung Diseases: A Pharmacological and Immunological Perspective*, pages 417–435, 2021.

[3] Josep Alòs, Carlos Ansótegui, and Eduard Torres. Interpretable decision trees through maxsat. *Artificial Intelligence Review*, 56(8):8303–8323, 2023.

[4] Julia Amann, Alessandro Blasimme, Effy Vayena, Dietmar Frey, Vince I Madai, and Precise4Q Consortium. Explainability for artificial intelligence in healthcare: a multidisciplinary perspective. *BMC medical informatics and decision making*, 20:1–9, 2020.

[5] P. Bellini, R. Mattolini, and P. Nesi. Temporal logics for real-time system specification. *ACM Comput. Surv.*, 32(1):12–42, mar 2000.

[6] Christian Bessiere, Emmanuel Hebrard, and Barry O'Sullivan. Minimising decision tree size as combinatorial optimisation. In *International Conference on Principles and Practice of Constraint Programming*, pages 173–187. Springer, 2009.

[7] Oliver Biggar and Mohammad Zamani. A framework for formal verification of behavior trees with linear temporal logic. *IEEE Robotics and Automation Letters*, 5(2):2341–2348, 2020.

[8] Christopher M. Bishop and Nasser M. Nasrabadi. *Pattern recognition and machine learning*, volume 1. springer New York, 2006.

[9] Franz Brauße, Zurab Khasidashvili, and Konstantin Korovin. Selecting stable safe configurations for systems modelled by neural networks with relu activation. In *2020 Formal Methods in Computer Aided Design*. IEEE, 2020.

[10] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[11] Rudy Bunel, Ilker Turkaslan, Philip H.S. Torr, Pushmeet Kohli, and M. Pawan Kumar. A unified view of piecewise linear neural network verification. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 4795–4804, Red Hook, NY, USA, 2018. Curran Associates Inc.

[12] Francesca Cairoli, Gianfranco Fenu, and Felice Andrea Pellegrino. Clinical decision support using colored petri nets: a case study on cancer infusion therapy. In *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*, pages 314–319, 2019.

[13] Christian Castaneda, Kip Nalley, Ciaran Mannion, Pritish Bhattacharyya, Patrick Blake, Andrew Pecora, Andre Goy, and K Stephen Suh. Clinical decision support systems for improving diagnostic accuracy and achieving precision medicine. *Journal of clinical bioinformatics*, 5:1–16, 2015.

[14] Manomita Chakraborty, Saroj Kumar Biswas, and Biswajit Purkayastha. Rule extraction from neural network using input data ranges recursively. *New Generation Computing*, 37:67–96, 2019.

[15] Yean Ru Chen, Tzu Fan Wang, Si-Han Chen, and Yi-Chun Kao. Empirical study on security verification and assessment of neural network accelerator. *Microprocessors and Microsystems*, 99:104845, 2023.

[16] May Y Choi and Christopher Ma. Making a big impact with small datasets using machine-learning approaches. *The Lancet Rheumatology*, 2(8):e451–e452, 2020.

[17] Daniel Clavel, Cristian Mahulea, and Manuel Silva. From healthcare system specifications to formal models. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 2344–2351, 2019.

[18] Arthur Clavière, Eric Asselin, Christophe Garion, and Claire Pagetti. Safety verification of neural network controlled systems. In *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pages 47–54, 2021.

[19] Darren Cofer, Ramachandra Sattigeri, Isaac Amundson, Junaid Babar, Saqib Hasan, Eric W. Smith, Karthik Nukala, Denis Osipychev, Matthew A. Moser, James L. Paunicka, Dragos D. Margineantu, Lucca Timmerman, and Jordan Q. Stringfield. Flight test of a collision avoidance neural network with run-time assurance. In *2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC)*, pages 1–10, 2022.

[20] Valency Oscar Colaco and Simin Nadjm-Tehrani. Formal verification of tree ensembles against real-world composite geometric perturbations. In *Workshop on Artificial Intelligence Safety 2023 (SafeAI 2023) co-located with the Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI 2023)*. CEUR-WS, 2023.

[21] Davide Corsi, Enrico Marchesini, and Alessandro Farinelli. Formal verification of neural networks for safety-critical tasks in deep reinforcement learning. In *Uncertainty in Artificial Intelligence*, pages 333–343. PMLR, 2021.

[22] Davide Corsi, Enrico Marchesini, Alessandro Farinelli, and Paolo Fiorini. Formal verification for safe deep reinforcement learning in trajectory generation. In *2020 Fourth IEEE International Conference on Robotic Computing (IRC)*, pages 352–359, 2020.

[23] Andressa C.M. da Silveira, Álvaro Sobrinho, Leandro Dias da Silva, Danilo F.S. Santos, Muhammad Nauman, and Angelo Perkusich. Harnessing coloured petri nets to

enhance machine learning: A simulation-based method for healthcare and beyond. *Simulation Modelling Practice and Theory*, page 103080, 2025.

[24] Matheus Soares de Araujo, Leandro Dias da Silva, Álvaro Sobrinho, Paulo Cunha, and Leonardo Montecchi. Reliability analysis of multi-parameter monitoring systems for intensive care units. *Reliability Engineering & System Safety*, 226:108638, 2022.

[25] Ana Carolina Costa de Oliveira, Emanuelle Silva de Mélo, Thayana Rose de Araújo Dantas, Ronei Marcos de Moraes, Messias Rafael Batista, and Marcelo Fernandes de Sousa. Aplicação do método jrip em variados bancos de dados. *Acta Scientia*, 3(2), 2021.

[26] Stefano Demarchi and Dario Guidotti. Counter-example guided abstract refinement for verification of neural networks. In *CPS Summer School, PhD Workshop*, 2022.

[27] Stefano Demarchi, Dario Guidotti, Andrea Pitto, and Armando Tacchella. Formal verification of neural networks: A case study about adaptive cruise control. In *Proceedings of the 36th ECMS International Conference on Modelling and Simulation*, pages 310–316, 2022.

[28] Lenardo Chaves e Silva, Álvaro Sobrinho, Thiago Cordeiro, Alan Pedro da Silva, Diego Dermeval, Leonardo Brandão Marques, Ig Ibert Bittencourt, Jário José dos Santos Júnior, Rafael Ferreira Melo, Carlos dos Santos Portela, Maurício Ronny de Almeida Souza, Rodrigo Lisbôa Pereira, Edson Koiti Kudo Yasojima, and Seiji Isotani. Assessing students' handwritten text productions: A two-decades literature review. *Expert Systems with Applications*, 250:123780, 2024.

[29] Ruediger Ehlers. Formal verification of piece-wise linear feed-forward neural networks. In *Automated Technology for Verification and Analysis: 15th International Symposium, ATVA 2017, Pune, India, October 3–6, 2017, Proceedings 15*, pages 269–286. Springer, 2017.

[30] Yizhak Yisrael Elboher, Elazar Cohen, and Guy Katz. Neural network verification using residual reasoning. In *International Conference on Software Engineering and Formal Methods*, pages 173–189. Springer, 2022.

[31] Yizhak Yisrael Elboher, Justin Gottschlich, and Guy Katz. An abstraction-based framework for neural network verification. In *Computer Aided Verification: 32nd International Conference, CAV 2020, Los Angeles, CA, USA, July 21–24, 2020, Proceedings, Part I 32*, pages 43–65. Springer, 2020.

[32] Javier Jesús Espinosa-Zúñiga. Aplicación de algoritmos random forest y xgboost en una base de solicitudes de tarjetas de crédito. *Ingeniería, investigación y tecnología*, 21(3), 2020.

[33] Katti Faceli, Ana Carolina Lorena, João Gama, and André Carlos Ponce de Leon Ferreira de Carvalho. Inteligência artificial: uma abordagem de aprendizado de máquina. 2011.

[34] Mahyar Fazlyab, Manfred Morari, and George J. Pappas. Safety verification and robustness analysis of neural networks via quadratic constraints and semidefinite programming. *IEEE Transactions on Automatic Control*, 67(1):1–15, 2022.

[35] Tássio Fernandes Costa, Alvaro Sobrinho, Lenardo Chaves e Silva, Leandro Dias da Silva, and Angelo Perkusich. Coloured petri nets-based modeling and validation of insulin infusion pump systems. *Applied Sciences*, 12(3), 2022.

[36] Nathan Fulton and André Platzer. Verifiably safe off-model reinforcement learning. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 413–430. Springer, 2019.

[37] Mohammad M Ghiasi and Sohrab Zendehboudi. Application of decision tree-based ensemble learning in the classification of breast cancer. *Computers in Biology and Medicine*, 128:104089, 2021.

[38] Mark L. Graber, Nancy Franklin, and Ruthanna Gordon. Diagnostic Error in Internal Medicine. *Archives of Internal Medicine*, 165(13):1493–1499, 07 2005.

[39] Gustavo Guerrero-Gomez, Faustino Moreno-Gamboa, et al. Design of a model for improving emergency room performance using a colored petri net. *EUREKA: Physics and Engineering*, (1):154–166, 2024.

[40] Dario Guidotti. Verification of neural networks for safety and security-critical domains. In *Proceedings of the International Conference of the Italian Association for Artificial Intelligence*, 2022.

[41] Vinícius Henrique Almeida Guimarães, Maísa de Oliveira-Leandro, Carolina Cassiano, Anna Laura Piantino Marques, Clara Motta, Ana Letícia Freitas-Silva, Marlos Aureliano Dias de Sousa, Luciano Alves Matias Silveira, Thiago César Pardi, Fernanda Castro Gazotto, Marcos Vinícius Silva, Virmondes Rodrigues Jr, Wellington Francisco Rodrigues, and Carlo Jose Freire Oliveira. Knowledge about covid-19 in brazil: Cross-sectional web-based study. *JMIR Public Health Surveill*, 7(1):e24756, Jan 2021.

[42] Longfei Han, Senlin Luo, Jianmin Yu, Limin Pan, and Songjing Chen. Rule extraction from support vector machines using ensemble learning approach: An application for diagnosis of diabetes. *IEEE Journal of Biomedical and Health Informatics*, 19(2):728–734, 2015.

[43] Hosein Hasanbeig, Daniel Kroening, and Alessandro Abate. Certified reinforcement learning with logic guidance. *Artificial Intelligence*, 322:103949, 2023.

[44] Mohammadhosein Hasanbeig, Daniel Kroening, and Alessandro Abate. Towards verifiable and safe model-free reinforcement learning. CEUR Workshop Proceedings, 2020.

[45] Nathan Hunt, Nathan Fulton, Sara Magliacane, Trong Nghia Hoang, Subhro Das, and Armando Solar-Lezama. Verifiably safe exploration for end-to-end reinforcement learning. In *Proceedings of the 24th International Conference on Hybrid Systems: Computation and Control*, HSCC '21, New York, NY, USA, 2021. Association for Computing Machinery.

[46] Ahmed Irfan, Kyle D. Julian, Haoze Wu, Clark Barrett, Mykel J. Kochenderfer, Baoluo Meng, and James Lopez. Towards verification of neural networks for small unmanned aircraft collision avoidance. In *2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC)*, pages 1–10, 2020.

[47] Radoslav Ivanov, Taylor J. Carpenter, James Weimer, Rajeev Alur, George J. Pappas, and Insup Lee. Case study: verifying the safety of an autonomous racing car with a neural network controller. In *Proceedings of the 23rd International Conference on Hybrid Systems: Computation and Control*, HSCC '20, New York, NY, USA, 2020. Association for Computing Machinery.

[48] Radoslav Ivanov, Taylor J. Carpenter, James Weimer, Rajeev Alur, George J. Pappas, and Insup Lee. Verifying the safety of autonomous systems with neural network controllers. *ACM Trans. Embed. Comput. Syst.*, 20(1), dec 2020.

[49] Nils Jansen, Bettina Könighofer, Sebastian Junges, Alex Serban, and Roderick Bloem. Safe reinforcement learning using probabilistic shields. In *31st International Conference on Concurrency Theory (CONCUR 2020)*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik, 2020.

[50] Mohd Javaid, Abid Haleem, Ravi Pratap Singh, Rajiv Suman, and Shanay Rab. Significance of machine learning in healthcare: Features, pillars and applications. *International Journal of Intelligent Networks*, 3:58–73, 2022.

[51] Kurt Jensen. Coloured petri nets and the invariant-method. *Theoretical computer science*, 14(3):317–336, 1981.

[52] Kurt Jensen. Kristensen, lm: Coloured petri nets: Modelling and validation of concurrent systems, 2009.

[53] Kurt Jensen and Lars M Kristensen. Colored petri nets: a graphical language for formal modeling and validation of concurrent systems. *Communications of the ACM*, 58(6):61–70, 2015.

[54] Nathanael Jo, Sina Aghaei, Jack Benson, Andres Gomez, and Phebe Vayanos. Learning optimal fair decision trees: Trade-offs between interpretability, fairness, and accuracy. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, pages 181–192, 2023.

[55] Charles Jones, Daniel C Castro, Fabio De Sousa Ribeiro, Ozan Oktay, Melissa McCradden, and Ben Glocker. A causal perspective on dataset bias in machine learning for medical imaging. *Nature Machine Intelligence*, pages 1–9, 2024.

[56] José Irineu Ferreira Júnior, Alvaro Sobrinho, Leandro Dias da Silva, Paulo Cunha, Thiago Cordeiro, Angelo Perkusich, and Antonio Marcus Nogueira Lima. A coloured petri nets-based system for validation of biomedical signal acquisition devices. *The Journal of Supercomputing*, pages 1–30, 2024.

[57] Fateh Kaakai, Said Hayat, and Abdellah El Moudni. A hybrid petri nets-based simulation model for evaluating the design of railway transit stations. *Simulation Modelling Practice and Theory*, 15(8):935–969, 2007.

[58] Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *Computer Aided Verification: 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part I 30*, pages 97–117. Springer, 2017.

[59] Guy Katz, Derek A Huang, Duligur Ibeling, Kyle Julian, Christopher Lazarus, Rachel Lim, Parth Shah, Shantanu Thakoor, Haoze Wu, Aleksandar Zeljić, et al. The marabou framework for verification and analysis of deep neural networks. In *Computer Aided Verification: 31st International Conference, CAV 2019, New York City, NY, USA, July 15-18, 2019, Proceedings, Part I 31*, pages 443–452. Springer, 2019.

[60] Sydney M Katz, Anthony L Corso, Christopher A Strong, and Mykel J Kochenderfer. Verification of image-based neural network controllers using generative models. *Journal of Aerospace Information Systems*, 19(9):574–584, 2022.

[61] Hojat Khosrowjerdi, Karl Meinke, and Andreas Rasmusson. Virtualized-fault injection testing: A machine learning approach. In *2018 IEEE 11th International Conference on Software Testing, Verification and Validation (ICST)*, pages 297–308, 2018.

[62] Sergey V Kovalchuk, Georgy D Kopanitsa, Ilia V Derevitskii, Georgy A Matveev, and Daria A Savitskaya. Three-stage intelligent support of clinical decision ma-

king for higher trust, validity, and explainability. *Journal of Biomedical Informatics*, 127:104013, 2022.

[63] Vrushali Y Kulkarni and Pradeep K Sinha. Pruning of random forest classifiers: A survey and future directions. In *2012 International Conference on Data Science & Engineering (ICDSE)*, pages 64–68, 2012.

[64] Taisa Kushner, Sriram Sankaranarayanan, and Marc Breton. Conformance verification for neural network models of glucose-insulin dynamics. In *Proceedings of the 23rd International Conference on Hybrid Systems: Computation and Control*, HSCC '20, New York, NY, USA, 2020. Association for Computing Machinery.

[65] Kim Larsen, Axel Legay, Gerrit Nolte, Maximilian Schlüter, Marielle Stoelinga, and Bernhard Steffen. Formal methods meet machine learning (f3ml). In Tiziana Margaria and Bernhard Steffen, editors, *Leveraging Applications of Formal Methods, Verification and Validation. Adaptation and Learning*, pages 393–405. Springer Nature Switzerland, 2022.

[66] Xiao Li, Zachary Serlin, Guang Yang, and Calin Belta. A formal methods approach to interpretable reinforcement learning for robotic planning. *Science Robotics*, 4(37):eaay6276, 2019.

[67] Zachary C Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, 2018.

[68] Tyler J. Loftus, Amanda C. Filiberto, Yanjun Li, Jeremy Balch, Allyson C. Cook, Patrick J. Tighe, Philip A. Efron, Gilbert R. Upchurch, Parisa Rashidi, Xiaolin Li, and Azra Bihorac. Decision analysis and reinforcement learning in surgical decision-making. *Surgery*, 168(2):253–266, 2020.

[69] Yuteng Lu, Weidi Sun, Guangdong Bai, and Meng Sun. Deepauto: A first step towards formal verification of deep learning systems (s). In *SEKE*, pages 172–176, 2021.

[70] Jianan Ma, Pengfei Yang, Jingyi Wang, Youcheng Sun, Cheng-Chao Huang, and Zhen Wang. Vere: Verification guided synthesis for repairing deep neural networks. In *Pro-*

ceedings of the IEEE/ACM 46th International Conference on Software Engineering, ICSE '24, New York, NY, USA, 2024. Association for Computing Machinery.

[71] Shucen Ma, Jianqi Shi, Yanhong Huang, Shengchao Qin, and Zhe Hou. Minimal-unsatisfiable-core-driven local explainability analysis for random forest. *Int. J. Softw. Informatics*, 12(4):355–376, 2022.

[72] Oded Z Maimon and Lior Rokach. *Data mining with decision trees: theory and applications*, volume 81. World scientific, 2014.

[73] Diego Manzanas Lopez, Taylor T Johnson, Stanley Bak, Hoang-Dung Tran, and Kerianne L Hobbs. Evaluation of neural network verification methods for air-to-air collision avoidance. *Journal of Air Transportation*, 31(1):1–17, 2023.

[74] Elias P Medeiros, Marcos R Machado, Emannuel Diego G de Freitas, Daniel S da Silva, and Renato William R de Souza. Applications of machine learning algorithms to support covid-19 diagnosis using x-rays data information. *Expert Systems with Applications*, 238:122029, 2024.

[75] Omar El Mellouki, Mohamed Ibn Khedher, and Mounim A. El-Yacoubi. Abstract layer for leakyrelu for neural network verification based on abstract interpretation. *IEEE Access*, 11:33401–33413, 2023.

[76] Samvid Mistry, Indranil Saha, and Swarnendu Biswas. An milp encoding for efficient verification of quantized deep neural networks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 41(11):4445–4456, 2022.

[77] Arnold S Monto, Stefan Gravenstein, Michael Elliott, Michael Colopy, and Jo Schweinle. Clinical signs and symptoms predicting influenza infection. *Archives of internal medicine*, 160(21):3243–3247, 2000.

[78] Pierre El Mqirmi, Francesco Belardinelli, and Borja G León. An abstraction-based method to check multi-agent deep reinforcement-learning behaviors. In *Proceedings of the International Conference on Autonomous Agents and MultiAgent Systems*, 2021.

[79] Mark Niklas Müller, Gleb Makarchuk, Gagandeep Singh, Markus Püschel, and Martin Vechev. Prima: general and precise neural network certification via scalable convex hull approximations. *Proc. ACM Program. Lang.*, 6(POPL), jan 2022.

[80] T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.

[81] W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Interpretable machine learning: definitions, methods, and applications. *arXiv preprint arXiv:1901.04592*, 2019.

[82] Muddasar Naeem, Syed Tahir Hussain Rizvi, and Antonio Coronato. A gentle introduction to reinforcement learning and its application in different fields. *IEEE Access*, 8:209320–209344, 2020.

[83] Muhammad Nauman, Nadeem Akhtar, Omar H. Alhazmi, Mustafa Hameed, Habib Ullah, and Nadia Khan. Improving the correctness of medical diagnostics based on machine learning with coloured petri nets. *IEEE Access*, 9:143434–143447, 2021.

[84] Muhammad Nauman, Nadeem Akhtar, Omar H. Alhazmi, Mustafa Hameed, Habib Ullah, and Nadia Khan. Improving the correctness of medical diagnostics based on machine learning with coloured petri nets. *IEEE Access*, 9:143434–143447, 2021.

[85] Muhammad Nauman, Nadeem Akhtar, Adi Alhudhaif, and Abdulrahman Alothaim. Guaranteeing correctness of machine learning based decision making at higher educational institutions. *IEEE Access*, 9:92864–92880, 2021.

[86] Muhammad Nauman, Nadeem Akhtar, Adi Alhudhaif, and Abdulrahman Alothaim. Guaranteeing correctness of machine learning based decision making at higher educational institutions. *IEEE access*, 9:92864–92880, 2021.

[87] David Olave-Rojas and Stefan Nickel. Modeling a pre-hospital emergency medical service using hybrid simulation and a machine learning approach. *Simulation Modelling Practice and Theory*, 109:102302, 2021.

[88] Thais Mayumi Oshiro, Pedro Santoro Perez, and José Augusto Baranauskas. How many trees in a random forest? In *Machine Learning and Data Mining in Pattern Recognition: 8th International Conference, MLDM 2012, Berlin, Germany, July 13-20, 2012. Proceedings 8*, pages 154–168. Springer, 2012.

[89] Colin Paterson, Haoze Wu, John Grese, Radu Calinescu, Corina S Păsăreanu, and Clark Barrett. Deepcert: Verification of contextually relevant robustness for neural network image classifiers. In *Computer Safety, Reliability, and Security: 40th International Conference, SAFECOMP 2021, York, UK, September 8–10, 2021, Proceedings 40*, pages 3–17. Springer, 2021.

[90] Kai Petersen, Sairam Vakkalanka, and Ludwik Kuzniarz. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64:1–18, 2015.

[91] Ameya Pore, Davide Corsi, Enrico Marchesini, Diego Dall'Alba, Alicia Casals, Alessandro Farinelli, and Paolo Fiorini. Safe reinforcement learning using formal verification for tissue retraction in autonomous robotic-assisted surgery. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4025–4031, 2021.

[92] Mohit Prashant and Arvind Easwaran. Pac-based formal verification for out-of-distribution data detection. In *2022 6th International Conference on System Reliability and Safety (ICSRS)*, pages 300–309, 2022.

[93] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1:81–106, 1986.

[94] Ramakrishnan Raman, Nikhil Gupta, and Yogananda Jeppu. Framework for formal verification of machine learning based complex system-of-systems. *INSIGHT*, 26(1):91–102, 2023.

[95] Ramakrishnan Raman, Nikhil Gupta, and Yogananda Jeppu. Framework for formal verification of machine learning based complex system-of-systems. *INSIGHT*, 26(1):91–102, 2023.

[96] Hao Ren, Sai Krishnan Chandrasekar, and Anitha Murugesan. Using quantifier elimination to enhance the safety assurance of deep neural networks. In *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*, pages 1–8. IEEE, 2019.

[97] Steven J Rigatti. Random forest. *Journal of Insurance Medicine*, 47(1):31–39, 2017.

[98] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215, 2019.

[99] Mercedes Ruiz, Elena Orta, and Juan Sánchez. A simulation-based approach for decision-support in healthcare processes. *Simulation Modelling Practice and Theory*, page 102983, 2024.

[100] Saima Safdar, Saad Zafar, Nadeem Zafar, and Naurin Farooq Khan. Machine learning based decision support systems (dss) for heart disease diagnosis: a review. *Artificial Intelligence Review*, 50(4):597–623, 2018.

[101] Shailendra Sahu and Babu M Mehtre. Network intrusion detection system using j48 decision tree. In *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 2023–2026. IEEE, 2015.

[102] Sincy Ann Saji, Shreyansh Agrawal, and Surinder Sood. Formal verification of deep neural networks in hardware. In *2022 IEEE Women in Technology Conference (WINTECHCON)*, pages 1–6, 2022.

[103] Ulices Santa Cruz and Yasser Shoukry. Nnlander-verif: A neural network formal verification framework for vision-based autonomous aircraft landing. In *NASA Formal Methods Symposium*, pages 213–230. Springer, 2022.

[104] Mucahid Mustafa Saritas and Ali Yasar. Performance analysis of ann and naive bayes classification algorithm for data classification. *International journal of intelligent systems and applications in engineering*, 7(2):88–91, 2019.

[105] Neil Savage. Breaking into the black box of artificial intelligence. *Nature*, 2022.

[106] Shruthi H. Shetty, Sumiksha Shetty, Chandra Singh, and Ashwath Rao. *Supervised Machine Learning: Algorithms and Applications*, chapter 1, pages 1–16. John Wiley & Sons, Ltd, 2022.

[107] KG Shojania, EC Burton, KM McDonald, and L Goldman. Autopsy as an outcome and performance measure: summary. In *AHRQ evidence report summaries*. Agency for Healthcare Research and Quality (US), 2002.

[108] Andressa C. M. da Silveira, Álvaro Sobrinho, Leandro Dias da Silva, Evandro de Barros Costa, Maria Eliete Pinheiro, and Angelo Perkusich. Exploring early prediction of chronic kidney disease using machine learning algorithms for small and imbalanced datasets. *Applied Sciences*, 12(7), 2022.

[109] Hardeep Singh, Ashley ND Meyer, and Eric J Thomas. The frequency of diagnostic errors in outpatient care: estimations from three large observational studies involving us adult populations. *BMJ quality & safety*, 23(9):727–731, 2014.

[110] Alvaro Sobrinho, Ially Almeida, Leandro Dias da Silva, Lenardo Chaves e Silva, Adriano Araújo, Tássio Fernandes Costa, and Angelo Perkusich. Coloured petri nets for abstract test generation in software engineering. *Software Testing, Verification and Reliability*, 33(2):e1837, 2023.

[111] Alvaro Sobrinho, Leandro Dias da Silva, Angelo Perkusich, Paulo Cunha, Thiago Cordeiro, and Antonio Marcus Nogueira Lima. Formal modeling of biomedical signal acquisition systems: source of evidence for certification. *Software & Systems Modeling*, 18:1467–1485, 2019.

[112] Alvaro Sobrinho, Leandro Dias da Silva, Maria Eliete Pinheiro, Paulo Cunha, Angelo Perkusich, and Leonardo Medeiros. Formal specification of a tool to aid the early dignosis of the chronic kidney disease. In *2015 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*, pages 173–178, 2015.

[113] Alvaro Sobrinho, Andressa C. M. Da S. Queiroz, Leandro Dias Da Silva, Evandro De Barros Costa, Maria Eliete Pinheiro, and Angelo Perkusich. Computer-aided di-

agnosis of chronic kidney disease in developing countries: A comparative analysis of machine learning techniques. *IEEE Access*, 8:25407–25419, 2020.

[114] Shiqi Sun, Yan Zhang, Xusheng Luo, Panagiotis Vlantis, Miroslav Pajic, and Michael M. Zavlanos. Formal verification of stochastic systems with relu neural network controllers. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6800–6806, 2022.

[115] Xiaowu Sun, Haitham Khedr, and Yasser Shoukry. Formal verification of neural network controlled autonomous systems. In *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control*, HSCC '19, page 147–156, New York, NY, USA, 2019. Association for Computing Machinery.

[116] Eric J Thomas, David M Studdert, Helen R Burstin, E John Orav, Timothy Zeena, Elliott J Williams, K Mason Howard, Paul C Weiler, and Troyen A Brennan. Incidence and types of adverse events and negligent care in utah and colorado. *Medical care*, 38(3):261–271, 2000.

[117] John Törnblom and Simin Nadjm-Tehrani. Formal verification of random forests in safety-critical applications. In *Formal Techniques for Safety-Critical Systems: 6th International Workshop, FTSCS 2018, Gold Coast, Australia, November 16, 2018, Revised Selected Papers 6*, pages 55–71. Springer, 2019.

[118] Hoang-Dung Tran, Diago Manzanas Lopez, Patrick Musau, Xiaodong Yang, Luan Viet Nguyen, Weiming Xiang, and Taylor T Johnson. Star-based reachability analysis of deep neural networks. In *Formal Methods–The Next 30 Years: Third World Congress, FM 2019, Porto, Portugal, October 7–11, 2019, Proceedings 3*, pages 670–686. Springer, 2019.

[119] John Törnblom and Simin Nadjm-Tehrani. Formal verification of input-output mappings of tree ensembles. *Science of Computer Programming*, 194:102450, 2020.

[120] Muhammad Usama, Junaid Qadir, Aunn Raza, Hunain Arif, Kok-lim Alvin Yau, Yehia Elkhatib, Amir Hussain, and Ala Al-Fuqaha. Unsupervised machine lear-

ning for networking: Techniques, applications and research challenges. *IEEE Access*, 7:65579–65615, 2019.

[121] Sudhir Varma and Richard Simon. Bias in error estimation when using cross-validation for model selection. *BMC bioinformatics*, 7(1):1–8, 2006.

[122] Baptiste Vasey, Myura Nagendran, Bruce Campbell, David A Clifton, Gary S Collins, Spiros Denaxas, Alastair K Denniston, Livia Faes, Bart Geerts, Mudathir Ibrahim, et al. Reporting guideline for the early stage clinical evaluation of decision support systems driven by artificial intelligence: Decide-ai. *bmj*, 377, 2022.

[123] Andreas Venzke and Spyros Chatzivasileiadis. Verification of neural network behaviour: Formal guarantees for power system applications. *IEEE Transactions on Smart Grid*, 12(1):383–397, 2021.

[124] Íris Viana dos Santos Santana, Andressa CM da Silveira, Álvaro Sobrinho, Lenardo Chaves e Silva, Leandro Dias da Silva, Danilo F S Santos, Edmar C Gurjão, and Angelo Perkusich. Classification models for covid-19 test prioritization in brazil: Machine learning approach. *J Med Internet Res*, 23(4):e27293, Apr 2021.

[125] Iris Viana dos Santos Santana, Alvaro Sobrinho, Leandro Dias da Silva, and Angelo Perkusich. Machine learning for covid-19 and influenza classification during coexisting outbreaks. *Applied Sciences*, 13(20), 2023.

[126] Yixuan Wang, Chao Huang, and Qi Zhu. Energy-efficient control adaptation with safety guarantees for learning-enabled cyber-physical systems. In *Proceedings of the 39th International Conference on Computer-Aided Design*, ICCAD '20, New York, NY, USA, 2020. Association for Computing Machinery.

[127] Yue Wang and Sai Ho Chung. Artificial intelligence in safety-critical systems: a systematic review. *Industrial Management & Data Systems*, 122(2):442–470, 2022.

[128] Zhilu Wang, Chao Huang, Yixuan Wang, Clara Hobbs, Samarjit Chakraborty, and Qi Zhu. Bounding perception neural network uncertainty for safe control of autonomous systems. In *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1745–1750, 2021.

[129] Michael Westergaard. Access/cpn 2.0: a high-level interface to coloured petri net models. In *Applications and Theory of Petri Nets: 32nd International Conference, PETRI NETS 2011, Newcastle, UK, June 20-24, 2011. Proceedings 32*, pages 328–337. Springer, 2011.

[130] Michael Westergaard and Lars Michael Kristensen. The access/cpn framework: A tool for interacting with the cpn tools simulator. In *International Conference on Applications and Theory of Petri Nets*, pages 313–322. Springer, 2009.

[131] Jim Woodcock, Peter Gorm Larsen, Juan Bicarregui, and John Fitzgerald. Formal methods: Practice and experience. *ACM Comput. Surv.*, 41(4), oct 2009.

[132] Qing Xu, Yicong Liu, Jian Pan, Jiawei Wang, Jianqiang Wang, and Keqiang Li. Reachability analysis plus satisfiability modulo theories: An adversary-proof control method for connected and autonomous vehicles. *IEEE Transactions on Industrial Electronics*, 70(3):2982–2992, 2023.

[133] Dhyan Chandra Yadav and Saurabh Pal. An experimental study of diversity of diabetes disease features by bagging and boosting ensemble method with rule based machine learning classifier algorithms. *SN Computer Science*, 2(1):50, 2021.

[134] Esen Yel, Taylor J. Carpenter, Carmelo Di Franco, Radoslav Ivanov, Yiannis Kantaros, Insup Lee, James Weimer, and Nicola Bezzo. Assured runtime monitoring and planning: Toward verification of neural networks for safe autonomous operations. *IEEE Robotics & Automation Magazine*, 27(2):102–116, 2020.

[135] Tom Zelazny, Haoze Wu, Clark Barrett, and Guy Katz. On optimizing back-substitution methods for neural network verification. In *2022 Formal Methods in Computer-Aided Design (FMCAD)*, pages 17–26. IEEE, 2022.

[136] Jie M. Zhang, Mark Harman, Lei Ma, and Yang Liu. Machine learning testing: Survey, landscapes and horizons. *IEEE Transactions on Software Engineering*, 48(1):1–36, 2022.

[137] Zhaodi Zhang, Jing Liu, Guanjun Liu, Jiacun Wang, and John Zhang. Robustness

verification of swish neural networks embedded in autonomous driving systems. *IEEE Transactions on Computational Social Systems*, 10(4):2041–2050, 2023.

[138] He Zhu, Zikang Xiong, Stephen Magill, and Suresh Jagannathan. An inductive synthesis framework for verifiable reinforcement learning. In *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*, PLDI 2019, page 686–701, New York, NY, USA, 2019. Association for Computing Machinery.

[139] Laura Zwaan, Martine de Bruijne, Cordula Wagner, Abel Thijs, Marleen Smits, Gerrit van der Wal, and Daniëlle RM Timmermans. Patient record review of the incidence, consequences, and causes of diagnostic adverse events. *Archives of internal medicine*, 170(12):1015–1021, 2010.