UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PRÓ-REITORIA DE GRADUAÇÃO
DEPARTAMENTO DE ENGENHARIAS E TECNOLOGIA
CURSO DE GRADUAÇÃO EM ENGENHARIA DE SOFTWARE

BRUNO VICTOR PAIVA DA SILVA

MODELO DE PERCEPTRON MULTICAMADAS UTILIZANDO COMPUTAÇÃO QUÂNTICA PARA ANALISAR RISCOS DE HIPERTENSÃO

PAU DOS FERROS
2025

BRUNO VICTOR PAIVA DA SILVA

MODELO DE PERCEPTRON MULTICAMADAS UTILIZANDO COMPUTAÇÃO QUÂNTICA PARA ANALISAR RISCOS DE HIPERTENSÃO

Monografia apresentada à Universidade Federal Rural do Semi-Árido como requisito para obtenção do título de Bacharel em Engenharia de Software.

Orientador: Prof. Dr. Walber José Adriano Silva - UFERSA

PAU DOS FERROS

2025

S586m Silva, Bruno Victor Paiva da.
Modelo de perceptron multicamadas utilizando computação quântica para analisar riscos de hipertensão / Bruno Victor Paiva da Silva. - 2025.
61 f. : il.

Orientador: Walber José Adriano Silva.
Monografia (graduação) - Universidade Federal Rural do Semi-árido, Curso de Engenharia de Software, 2025.

1. Aprendizado profundo. 2. Computação quântica. 3. Pressão arterial. 4. Aprendizado de máquina. 5. IA explicável. I. Silva, Walber José Adriano, orient. II. Título.

Ficha catalográfica elaborada por sistema gerador automático em conformidade com AACR2 e os dados fornecidos pelo) autor(a).
Biblioteca Campus Pau dos Ferros
Bibliotecária: Erlanda Maria Lopes da Silva
CRB: 15/966

O serviço de Geração Automática de Ficha Catalográfica para Trabalhos de Conclusão de Curso (TCC´s) foi desenvolvido pelo Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (USP) e gentilmente cedido para o Sistema de Bibliotecas da Universidade Federal Rural do Semi-Árido (SISBI-UFERSA), sendo customizado pela Superintendência de Tecnologia da Informação e Comunicação (SUTIC) sob orientação dos bibliotecários da instituição para ser adaptado às necessidades dos alunos dos Cursos de Graduação e Programas de Pós-Graduação da Universidade.

BRUNO VICTOR PAIVA DA SILVA

MODELO DE PERCEPTRON MULTICAMADAS UTILIZANDO COMPUTAÇÃO QUÂNTICA PARA ANALISAR RISCOS DE HIPERTENSÃO

Monografia apresentada à Universidade Federal Rural do Semi-Árido como requisito para obtenção do título de Bacharel em Engenharia de Software.

Defendida em: 11 de dezembro de 2025

BANCA EXAMINADORA

Walber José Adriano Silva, Prof. Dr. (UFERSA)
Presidente

Francisco Ernandes Matos Costa, Prof. Dr. (UFERSA)
Membro Examinador

João Paulo de Souza Medeiros, Prof. Dr. (UFRN)
Membro Examinador

Dedico este trabalho a minha mãe, pelo apoio de sempre, e a todos que de alguma forma colaboraram para a conclusão desta pesquisa.

AGRADECIMENTOS

Agradeço, em primeiro lugar, a Deus por estar presente em todos os momentos da minha vida, me dando forças e coragem para que eu possa dar o melhor de mim a cada dia.

Agradeço à minha família, em especial à minha mãe, Maria Helena Farias de Paiva, e ao meu irmão, Thiago Vinicius Paiva da Silva, por estarem comigo desde o início da minha jornada, me apoiando e dando todo suporte necessário, ajudando-me a superar todos os desafios.

Agradeço extremamente ao meu orientador, professor e amigo Dr. Walber José Adriano Silva, pela orientação, conselhos, amizade e por todas as oportunidades que me deu para que eu pudesse crescer em minha carreira acadêmica e por todo o conhecimento que proporcionou.

Agradeço aos professores, Dr. Ítalo Augusto Souza de Assis, Dr. Alysson Filgueira Milanez, Drª. Rosana Cibely Batista Rego, Drª Wagna Maquis Cardoso de Melo Gonçalves e Me. José Ferdinandy Silva Chagas, pelo companheirismo, pelas oportunidades e pelo conhecimento.

Por fim, agradeço às amizades que cultivei em toda essa jornada, em especial aos colegas Felipe Hidequel, Gabriel, Abner, Tiago, Isabela, Heloíze, Fernanda, Lívia, Leonardo e Geísa. E ainda dentro das amizades, também agradeço a todos os "Covardes", Kennedy, Augusto, Jhoan, Artur, Heitor, Vladimir, Nattan Ferreira Lopes, Pedro Lucas, Pedro Andrade, Renato, Vinicius, Elton e Pedro Raulino.

EPÍGRAFE

"*stay hungry, stay foolish.*"

(Steve Jobs)

RESUMO

A hipertensão é uma condição em que a pressão do sangue dentro das artérias permanece elevada por longos períodos. Essa condição é subdiagnosticada, principalmente porque muitos indivíduos não apresentam sintomas em seu estágio inicial, no qual o diagnóstico aparece após o surgimento de complicações, em que o dano ao sistema cardiovascular já está avançado. Assim, há a necessidade de um diagnóstico antecipado, capaz de identificar a doença antes que ela progrida. O presente trabalho explora essa lacuna buscando métodos mais eficientes com o uso de *deep learning* e computação quântica capazes de realizar o diagnóstico preliminar dessa condição por meio de variáveis clínicas de influência. O principal objetivo é o desenvolvimento de um modelo de *deep learning* clássico e de um segundo modelo com camadas internamente híbridas e comparar por meio de métricas de precisão entre os dois modelos, além de realizar a análise das variáveis de influência. Em seguida, foram realizados testes para avaliar o número de acertos dos modelos, precisão, sensibilidade, *F1-score* e tempo de treinamento. Portanto, os resultados obtidos indicam que os modelos desenvolvidos oferecem um recurso para a predição de riscos de hipertensão. Em particular, o modelo híbrido demonstrou potencial, considerando as perspectivas de avanço da computação quântica, para ampliar ainda mais sua eficácia.

Palavras-chave: aprendizado profundo; computação quântica; pressão arterial; aprendizado de máquina; IA explicável.

ABSTRACT

Hypertension is a condition in which blood pressure within the arteries remains elevated for long periods. This condition is often under-detected, mainly because many individuals do not present symptoms in its early stages, leading to diagnosis only after complications arise, when cardiovascular damage is already advanced. Thus, there is a clear need for an early diagnosis capable of identifying the disease before it progresses. This study aims to address this gap by investigating more efficient methods based on deep learning and quantum computing, capable of performing a preliminary diagnosis of this condition using influential clinical variables. The main objective is to develop a classical deep learning model and a second model with internally hybrid layers, comparing them through evaluation metrics and analyzing the most influential variables. Subsequently, tests were conducted to assess the performance of the models in terms of accuracy, precision, sensitivity, F1-score, and training time. The results indicate that the developed models provide a resource for predicting the risk of hypertension.

Keywords: deep learning; quantum computing; blood pressure; machine learning; explainable XAI.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE CÓDIGOS-FONTE

LISTA DE ABREVIATURAS E SIGLAS

AI	*Artificial Intelligence*
ANN	*Artificial Neural Network*
CNN	*Convolucional Neural Network*
CRISPR	*Clustered Regularly Interspaced Short Palindromic Repeats*
DL	*Deep Learning*
DNA	*Deoxyribonucleic Acid*
DNN	*Deep Neural Network*
ECG	Eletrocardiograma
FN	*False Negative*
FP	*False Positive*
GPU	Unidade de Processamento Gráfico
IMC	Índice de Massa Corporal
ML	*Machine Learning*
MLP	*Multilayer Perceptron*
PPG	Foto Pletismograma
PPV	*Positive Predictive Value*
ReLU	*Rectified Linear Unit*
RNN	*Recurrent Neural Network*
SHAP	*SHapley Additive exPlanations*
SMOTE	*Synthetic Minority Over-sampling Technique*
SVM	*Support Vector Machines*
tanh	*Tangente Hiperbólica*
TN	*True Negative*
TP	*True Positive*
XAI	Inteligência Artificial Explicável

LISTA DE SÍMBOLOS

$\hat{y}$	Símbolo de predição.
x_i	Variável de entrada de índice i.
w_i	Peso ou coeficiente associado à variável x_i.
α	Parâmetro escalar.
β	Parâmetro escalar.
$\|\psi\rangle$	Vetor de estado (ket) no formalismo de Dirac.
$\langle\psi\|$	Dual (bra) correspondente ao ket $\|\psi\rangle$.
$\|\psi_1\rangle, \|\psi_2\rangle$	Estados-base do sistema.
c_1, c_2	Amplitudes complexas associadas aos estados-base.

SUMÁRIO

1 INTRODUÇÃO

A hipertensão é uma condição crônica definida pela elevação da pressão arterial (Krzeminska *et al.*, 2022). Tornando-se mais comum com o envelhecimento, essa condição é multifatorial, tendo o estresse oxidativo como uma de suas principais causas (Donato; Machin; Lesniewski, 2018). Segundo Mills, Stefanescu e He (2020), a hipertensão é a principal responsável por doenças cardiovasculares e mortes prematuras em todo o mundo. A progressão da doença é habitualmente assintomática, o que resulta em um diagnóstico tardio, ocorrendo, em muitos casos, somente após o surgimento de complicações em órgãos-alvo.

O fato da doença começar antes de ser diagnosticada mostra que é preciso identifica-lá em seus estágios iniciais. Detectar o problema cedo permite iniciar tratamentos para controlar a doença e reduzir as taxas de mortalidade.

A *Artificial Intelligence* (AI) tem emergido como uma tecnologia que oferece a capacidade de analisar grandes volumes de dados na área da saúde com o pretexto de identificar padrões complexos e preditivos.

Nesse contexto, o *Machine Learning* (ML), um dos campos da AI, oferece um conjunto de técnicas para construção de modelos preditivos. Dentro do ML, as *Artificial Neural Network*s (ANNs) representam uma classe de modelos computacionais que tentam mimetizar o funcionamento do cérebro humano.

As arquiteturas das ANNs, tem sua especialidade conforme o tipo de dados e tarefas. Notadamente, as *Convolucional Neural Network*s (CNNs) destacam-se pelo seu desempenho no processamento de dados espaciais, especialmente imagens. Enquanto as *Recurrent Neural Network*s (RNNs) são projetadas para dados sequenciais, como textos.

Dentre as arquiteturas, a *Multilayer Perceptron* (MLP) destaca-se por ser um modelo de propósito geral e por ser geralmente aplicada para problemas de classificação e regressão baseados em dados estruturados. Além de serem reconhecidas como uma das arquiteturas de maior versatilidade em termos de uso (Gomide, 2012).

As MLPs têm sido empregados em diversas aplicações de diagnóstico. No entanto, a complexidade dos dados de saúde e a necessidade de processamento mais robusto justificam a busca por abordagens mais avançadas. A presente pesquisa propõe o desenvolvimento de uma rede neural híbrida que integra o poder da computação quântica para otimizar a detecção de riscos de hipertensão, buscando superar as limitações de modelos puramente clássicos.

1.1 Problematização

Apesar dos avanços recentes na área da AI aplicada à saúde, ainda existem limitações significativas na detecção de riscos de hipertensão (Ali *et al.*, 2022). Modelos tradicionais de redes neurais, embora eficazes, apresentam restrições quanto ao desempenho em novos conjuntos de dados e à alta demanda computacional. Além disso, observa-se que a aplicação de técnicas híbridas, que integrem elementos da computação quântica, permanece pouco explorada na literatura científica.

Diante disso, surge a questão central: como a integração de camadas quânticas em arquiteturas clássicas pode ampliar a capacidade de predição de riscos de hipertensão em comparação a arquitetura de *Deep Learning* (DL) ? Este trabalho busca, portanto, responder a esse questionamento, desenvolvendo um modelo clássico de rede neural, além de um modelo híbrido-clássico utilizando conceitos da computação quântica para aprimoramento e análise do modelo, através da adoção da arquitetura MLP, na qual é uma das arquiteturas mais simples da ANN (Geron, 2021).

1.2 Objetivo

O presente trabalho tem como objetivo geral desenvolver um modelo de rede neural baseado em uma arquitetura MLP de multicamadas para a detecção de riscos de hipertensão, incluindo a proposta de uma variante híbrida com camadas quânticas.

1.3 Objetivos Específicos

A fim de atender ao objetivo geral, definiram-se os seguintes objetivos específicos:

- Implementar a arquitetura de rede neural tradicional utilizando MLP;
- Integrar camadas quânticas ao modelo, gerando uma versão híbrida quântico-clássica;
- Avaliar e comparar o desempenho do modelo tradicional e do híbrido em termos de métricas relevantes;
- Conduzir uma análise comparativa dos modelos desenvolvidos em relação às arquiteturas avançadas de DL.

1.4 Justificativa

A hipertensão arterial é uma condição crônica de grande relevância para a saúde pública, em que é considerada como um dos principais fatores de risco para doenças cardiovasculares. Métodos de diagnóstico precoce podem auxiliar na redução de complicações e no planejamento de estratégias de prevenção. Nesse contexto, modelos baseados em redes neurais têm demonstrado desempenho promissor, porém ainda existem limitações quanto à sua capacidade de prever corretamente dados desconhecidos (Ali *et al.*, 2022).

No entanto, a computação quântica surge como uma área em desenvolvimento, oferecendo novas possibilidades de aceleração de cálculos e de exploração de arquiteturas híbridas em aprendizado de máquina. Assim, a proposta deste trabalho justifica-se pela necessidade de investigar alternativas que combinem técnicas clássicas e quânticas, avaliando seu potencial para melhorar a detecção de riscos de hipertensão no qual os resultados obtidos contribuem para o avanço de soluções tecnológicas voltadas à saúde.

2 REVISÃO TEÓRICA

Neste capítulo apresentamos um conteúdo base sobre hipertensão e seus principais riscos na Seção 2.1. Na Seção 2.2, apresentam-se os principais conceitos e aplicações presentes em uma rede neural artificial, além de técnicas e conceitos para aprimoramento de modelos em arquiteturas de MLPs. Na Seção 2.3 apresentamos sobre computação quântica e seus principais conceitos, como a superposição, emaranhamento, *qubits* e entre outros.

2.1 Hipertensão

A hipertensão, também conhecida como pressão arterial elevada, é uma condição em que a pressão do sangue nas artérias se mantém em níveis elevados. Segundo, Williams *et al.* (2018) a hipertensão pode ser classificada em três estágios conforme o nível de pressão arterial, presença de fatores de risco cardiovasculares, danos em órgãos mediados pela hipertensão ou comorbidades.

O primeiro estágio, denominado não complicado, corresponde a pacientes hipertensos sem lesões de órgãos-alvo e sem doenças associadas. O segundo estágio, denominado doença assintomática, caracteriza-se pela presença de lesão em órgãos-alvo, doença renal crônica em estágio 3 ou *diabetes mellitus*[1] sem lesão de órgãos. Já o terceiro estágio, considerado o mais avançado, é chamado de doença estabelecida e está relacionado à presença de doença cardiovascular estabelecida, doença renal crônica em estágio igual ou superior a 4, ou *diabetes mellitus* com lesão de órgãos-alvo.

Considerando essas definições, a Tabela 1 apresenta a estratificação de risco, a qual permite correlacionar os níveis de pressão arterial com a presença de fatores de risco adicionais, lesão de órgãos-alvo e doenças associadas.

Tabela 1 – Classificação da pressão arterial em adultos mmHg.

Classificação	**PAS (mmHg)**	**PAD (mmHg)**
Ótima	< 120	< 80
Normal	120–129	80–84
Normal alta (Pré-hipertensão)	130–139	85–89
Hipertensão Grau 1	140–159	90–99
Hipertensão Grau 2	160–179	100–109
Hipertensão Grau 3	≥ 180	≥ 110
Hipertensão Sistólica Isolada (HSI)	≥ 140	< 90

[1] Disponível em: https://www.gov.br/saude/pt-br/assuntos/saude-de-a-a-z/d/diabetes

2.2 Redes Neurais Artificiais

Na classificação de risco, os níveis de pressão arterial estão associados a diferentes fatores adicionais, como a idade mais avançada, consumo excessivo de sal, sedentarismo e entre outros. Para realizar uma análise agregada a esses fatores podem ser aplicados métodos computacionais, como as Redes Neurais Artificiais.

A primeira arquitetura de uma rede neural artificial foi proposta em 1943 por McCulloch e Pitts, configurando-se como um modelo computacional simplificado destinado a representar o funcionamento cooperativo dos neurônios biológicos no cérebro de animais. Esse modelo buscava demonstrar a capacidade dos neurônios artificiais de realizar cálculos complexos a partir da lógica proposicional (McCulloch; Pitts, 1943).

A MLP é uma das arquiteturas mais simples da ANN (Geron, 2021). A MLP é composta de uma camada de entrada, uma ou mais camadas ocultas, e uma camada final, chamada de camada de saída. As camadas mais próximas da camada de entrada são denominadas camadas inferiores e as camadas próximas à camada de saída são chamadas de camadas superiores. Com exceção da camada de saída, todas as outras camadas estão conectadas.

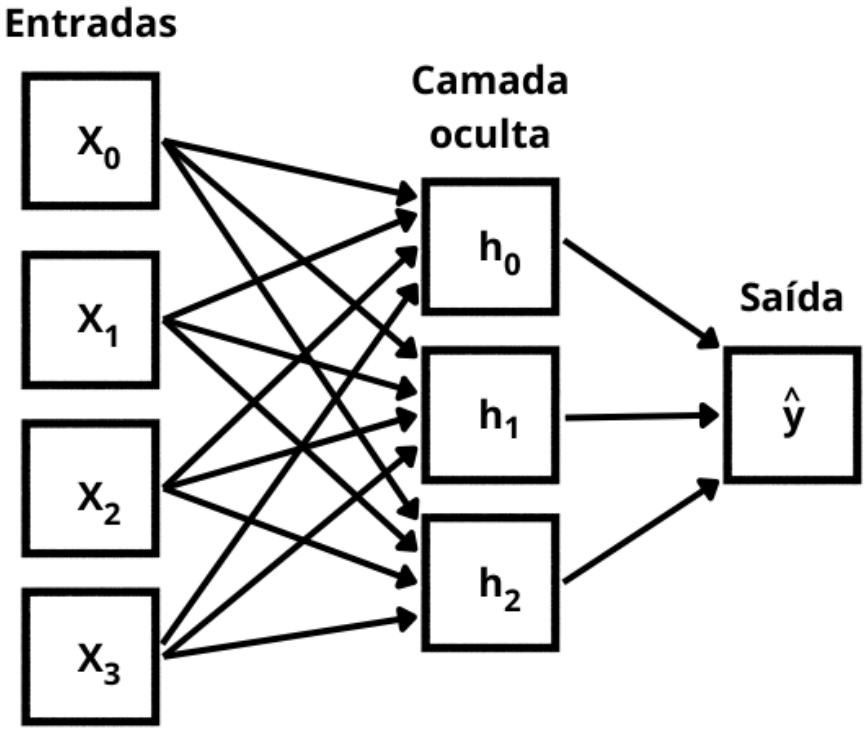
As MLPs são generalizações de modelos lineares que realizam múltiplas etapas de processamento para chegar a uma decisão (Müller; Guido, 2016), no qual a predição feita por uma regressão linear é apresentada na Equação 2.1.

$$\hat{y} = w_0 \cdot x_0 + w_1 \cdot x_1 + \cdots + w_p \cdot x_p + b \tag{2.1}$$

O $\hat{y}$ é uma soma ponderada das características de entrada x_0 a x_p, ponderadas pelos coeficientes aprendidos w_0 a w_p. Em uma MLP, o processo de cálculos de somas ponderadas é repetido múltiplas vezes, primeiro computando unidades ocultas que representam uma etapa intermediária de processamento, em que são novamente combinadas usando somas ponderadas para gerar o resultado, conforme ilustrado na Figura 1.

Na Figura 1, x_0, x_1, x_2 e x_3 representam os dados de entrada para a rede MLP, em que são ligados a camada oculta retratada por h_0, h_1 e h_2, no qual as unidades da camada oculta estão conectadas a um único nó de saída. Quando uma RNA possui uma pilha profunda de camadas ocultas, é chamada de *Deep Neural Network* (DNN). As DNNs pertencem à área que estuda modelos contendo pilhas profundas de cálculos.

Figura 1 – Ilustração de uma *perceptron* com uma única camada oculta



Fonte: Adaptado de Müller e Guido (2016).

No entanto, a capacidade de modelos complexos, como as DNNs, introduz um desafio comum no treinamento de redes neurais: o sobreajuste (*Overfitting*). Este fenômeno ocorre, pois o algoritmo é treinado em um conjunto de dados, mas aplicado em novos dados, os dados de testes. Segundo Dietterich (1995), se o modelo se ajusta excessivamente aos dados de treinamento, ele acaba aprendendo ruídos ou peculiaridades específicas, em vez de regras gerais.

2.2.1 Técnicas para Mitigação do *Overfitting*

Para reduzir o problema do sobreajuste, diversas técnicas são propostas, como a parada antecipada (*Early Stopping*), redução de rede (*Network Reduction*), expansão de dados (*Data Expansion*) e a regularização (*Regularization*) (Ying, 2019).

A técnica de *Early Stopping* é uma forma de regularização, na qual consiste em monitorar o desempenho do modelo em um conjunto de dados de validação ao longo das épocas de treinamento. O processo é interrompido quando a função de perda para de apresentar melhorias e começa a se degradar, ou seja, após atingir um ponto mínimo (Geron, 2021).

A *Network Reduction* visa reduzir o ruído. Com isso, a técnica de *Pruning* é proposta para reduzir o tamanho dos classificadores finais na aprendizagem relacional, especialmente na

aprendizagem de árvores de decisão. A *pruning* é uma teoria usada para reduzir a complexidade da classificação, eliminando dados menos significativos ou irrelevantes e, finalmente, para prevenir o sobreajuste e melhorar a acurácia da classificação (Ying, 2019).

A técnica de *Data Expansion* diz a respeito da quantidade e da qualidade do conjunto de dados de treinamento, especialmente no contexto de aprendizado supervisionado: envolve modelar a relação entre as medições dos dados com os rótulos associados (VanderPlas, 2016). O número de amostras deve ser proporcional ao número de parâmetros, quanto mais complexo for o modelo, mais parâmetros precisarão ser ajustados.

Das técnicas presentes no *Regularization*, o *dropout* é uma das mais populares e eficazes para combater o *overfitting* (Ying, 2019). A função do *dropout* é desativar aleatoriamente algumas unidades de neurônios e suas conexões durante o treinamento, para evitar que tenha unidades dependentes uma das outras. Observa-se, na Equação 2.2, a fórmula que descreve o *dropout*.

$$\mathbf{a}_{\text{dropout}} = \begin{cases} \frac{a}{p}, & \text{com probabilidade } p \\ 0, & \text{com probabilidade } (1-p) \end{cases} \tag{2.2}$$

Na Equação 2.2, $\mathbf{a}_{\text{dropout}}$ representa a saída após a aplicação do *dropout*, no qual a é a ativação do neurônio antes da aplicação do *dropout*. A variável p indica a probabilidade de manter um neurônio ativo durante o treinamento, ou seja, a fração de neurônios que permanecem ativados. Com probabilidade p, a ativação é ajustada proporcionalmente a esse valor para preservar a média esperada da saída, já com probabilidade 1-p, a ativação é zerada, o que desativa temporariamente o neurônio durante o treinamento.

2.3 Computação Quântica

A computação quântica consiste em um estudo de fenômenos da mecânica quântica para o processamento de informações (Nielsen; Chuang, 2010), no qual computadores de arquitetura clássica se mostram limitados. Alguns princípios fundamentais para a computação quântica incluem a superposição de estado e o emaranhamento quântico. A superposição na física quântica acontece quando um sistema quântico pode existir em vários estados ao mesmo tempo, antes de ser observado ou medido, isso implica que um objeto pode estar em uma combinação linear

de múltiplos estados possíveis simultaneamente (Shankar, 2012). A notação de Dirac para superposição quântica é observada na Equação 2.3 (Dirac, 1981).

$$|\psi\rangle = c_1|\psi_1\rangle + c_2|\psi_2\rangle \tag{2.3}$$

Na Equação 2.3, o símbolo de ψ (*psi*) é conhecido como *ket* representa o estado quântico de um sistema. Os coeficientes c_1 e c_2 representam as amplitudes de probabilidades complexas, ou seja, a probabilidade de o sistema estar em cada um dos estados. Os estados $|\psi_1\rangle$ e $|\psi_2\rangle$ são estados-bases possíveis de 0 e 1.

No emaranhamento, dois ou mais sistemas quânticos tornam-se correlacionados, de modo que o estado de um deles está diretamente relacionado ao estado do outro, independentemente da distância que os separa (Preskill, 2018). Por meio da superposição e do emaranhamento, a computação quântica pode realizar cálculos simultâneos em múltiplos estados, em que a capacidade de processamento é exponencialmente maior do que os computadores clássicos para determinados tipos de problema, como problemas de fatoração inteira relacionadas a criptografia, buscas não estruturadas e problemas de otimização combinatória (Harrow; Montanaro, 2017; Nielsen; Chuang, 2010). O estado emaranhado característico para um sistema de duas partículas é o estado de Bell, descrito conforme a Equação 2.4.

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle_A|1\rangle_B + |1\rangle_A|0\rangle_B) \tag{2.4}$$

No estado de Bell, as partículas *A* e *B* indicam um estado de superposição, no qual se *A* for medida no estado $|0\rangle$, então *B* será encontrada no estado $|1\rangle$. Já $|0\rangle$ e $|1\rangle$ são os estados computacionais, também conhecidos como estados bases. O fator $\frac{1}{\sqrt{2}}$ é um coeficiente de normalização, garantindo que a soma das probabilidades do sistema seja 1.

Enquanto o *bit* é a unidade básica de informação na computação clássica, podendo assumir apenas um dos dois estados discretos, 0 ou 1, o *qubit* é a unidade básica de informação na computação quântica, sendo capaz de existir em uma superposição dos estados $|0\rangle$ e $|1\rangle$ simultaneamente. A diferença entre *bits* e *qubits* é que um *qubit* pode estar em um estado diferente de $|0\rangle$ ou $|1\rangle$. A representação de um *qubit* é apresentada na Equação 2.3.

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{2.5}$$

A interação dos *qubits* com o ambiente externo resulta na perda da informação quântica devido a ruídos e perturbações, gerando uma decoerência quântica. Os *qubits* são controlados por meio das portas quânticas, responsáveis por modificar seus estados a fim de executar operações computacionais. Por meio delas, é possível gerar, transformar e medir estados quânticos. Algumas das portas lógicas mais conhecidas são as de *Hadamard, CNOT, Pauli -X, Pauli -Y e Pauli -Z.*

As portas lógicas que operam em mais de 1 *qubit* exploram o poder da computação quântica, permitindo a criação de estados emaranhados e a implementação de interações complexas entre *qubits*. A realização das operações simultâneas em inúmeros estados, em que atuam em todos os valores da superposição ao mesmo tempo, é denominado de paralelismo quântico.

2.4 Métricas de Desempenho

Para avaliar os resultados de modelos de aprendizado de máquina ou de aprendizado profundo, métricas permitem quantificar o desempenho de um modelo (Varoquaux; Colliot, 2023). Nas tarefas de classificação, os resultados podem ser resumidos em uma matriz chamada matriz de confusão, apresentada na Tabela 2.

		Rótulo Verdadeiro	
		Positivo (D^+)	**Negativo** (D^-)
Rótulo Previsto		**TP**	**FP**
		(Verdadeiro Positivo)	(Falso Positivo)
		FN	**TN**
		(Falso Negativo)	(Verdadeiro Negativo)

Tabela 2 – Matriz de confusão para classificação binária.

No caso de uma classificação binária, a matriz de confusão divide as amostras de teste em quatro categorias, de acordo com seus rótulos verdadeiros e previstos. Na Tabela 2, os *True Positive* (TP) são amostras corretamente classificadas como positivas. Os *False Positive* (FP) são amostras incorretamente classificadas como positivas. Já os *True Negative* (TN) são amostras incorretamente classificadas como positivas. Por fim, *False Negative* (FN) são amostras incorretamente classificadas como negativas. Para avaliar as informações contidas na matriz de confusão, algumas métricas, como a acurácia expressa na Equação 2.6:

$$\text{Acurácia} = \frac{TP+TN}{TP+TN+FP+FN} \tag{2.6}$$

A acurácia é o número de previsões corretas (TP e TN) dividido pelo número total de amostras, ou seja, a soma de todas as entradas da matriz de confusão. Outra métrica, a precisão (Equação 2.7) mede quantas das amostras previstas como positivas são realmente positivas.

$$\text{Precisão} = \frac{TP}{TP+FP} \tag{2.7}$$

A precisão é usada como uma métrica de desempenho quando o objetivo é limitar o número de falsos positivos. A precisão também é conhecida como *Positive Predictive Value* (PPV). Por outro lado, a sensibilidade, conhecida como *recall* (Equação 2.8) mede quantas das amostras positivas reais são corretamente identificadas pelo modelo:

$$\text{Sensibilidade} = \frac{TP}{TP+FN} \tag{2.8}$$

A sensibilidade é usada como métrica de desempenho quando é necessário identificar todas as amostras positivas, ou seja, quando é importante evitar falsos negativos. A precisão e a sensibilidade são as métricas mais usadas para classificação binária (Müller; Guido, 2016). Uma maneira de resumi-lá sem uma única métrica é o *F1-score* (Equação 2.9), definido como a média harmônica entre precisão e *recall*:

$$F = 2 \cdot \frac{\text{Precisão} \cdot \text{recall}}{\text{Precisão} + \text{recall}} \tag{2.9}$$

3 TRABALHOS RELACIONADOS

Este capítulo apresenta publicações científicas relacionadas ao tema desta pesquisa. Na Seção 3.1, Zhao *et al.* (2021) realizam a predição de riscos de hipertensão utilizando modelos de aprendizado de máquina e comparando o seu desempenho. Já na Seção 3.2, Marin e Goga (2019) realiza a predição com uma base de dados para diferentes grupos utilizando quatro algoritmos de aprendizado de máquina.

Na Seção 3.3, os trabalhos de Dond (2025) e Gong, Chen e Ding (2023) exploram técnicas de aprendizado de máquina relacionadas a medições de pressão arterial e hábitos de vida. Já na Seção 3.4, Gabrich, Freitas e Souza (2024) aplicam conceitos da mecânica quântica para construção de modelos inteligentes capazes de interferir em dados genéticos.

3.1 Modelos de aprendizado de máquina para predição de hipertensão

O estudo de Zhao *et al.* (2021) investiga a predição do risco de hipertensão, uma condição crônica amplamente disseminada, por meio de modelos de aprendizado de máquina capazes de operar com fatores de risco simples e facilmente coletáveis. Utilizando um conjunto de dados composto por 29.700 amostras obtidas em exames físicos, os autores selecionaram inicialmente os principais fatores de risco por meio de regressão logística univariada.

Com essas variáveis selecionadas, quatro algoritmos: *Random Forest*, *CatBoost*, rede neural MLP e Regressão Logística foram otimizados por validação cruzada *10-fold* para ajuste dos hiperparâmetros. O desempenho final foi avaliado a partir de métricas como a área sob curva que mede a capacidade geral do modelo de separar as classes, acurácia, sensibilidade e especificidade que mede a capacidade do modelo de identificar corretamente os exemplos negativos no conjunto de teste. Entre os modelos testados, o *Random Forest* apresentou os melhores resultados apresentados na Tabela 3.

Acurácia (%)	Sensibilidade (%)	Especificidade (%)	Área sob curva (%)
82%	83%	81%	92%

Tabela 3 – Métricas de desempenho do estudo do modelo clássico de aprendizado de máquina extraídas de 29.700 amostras deitas pelo trabalho de Zhao *et al.* (2021).

3.2 Detecção hipertensão com classificadores de aprendizado de máquina em diferentes grupos populacionais

Já o trabalho de Marin e Goga (2019), teve como objetivo usar aprendizado de máquina para identificar hipertensão em diferentes grupos (população geral, pacientes hipertensos e gestantes com ou sem pré-eclâmpsia). Esse trabalho utilizou bases de dados públicas da Universidade de *Massachusetts Amherst*[2] e do *National Health and Nutrition Survey*[3].

De modo a encontrar os resultados, foram empregados quatro algoritmos de *Machine Learning*. O primeiro, *Gaussian Naive Bayes*, é um classificador probabilístico. O segundo e o terceiro, respectivamente Regressão Logística e *Random Forest*, baseiam-se na votação de um conjunto de árvores de decisão. Por fim, o *Support Vector Machines* (SVM) atua encontrando um separador entre classes de dados.

Os resultados mostraram que os modelos identificaram corretamente os indivíduos sem hipertensão, mas acertaram apenas aproximadamente metade dos casos em que a hipertensão estava presente, indicando um problema de sensibilidade nas detecções positivas. No entanto, os autores reforçam que a detecção automática de parâmetros de saúde fora do padrão, por meio de técnicas de aprendizado de máquina, é uma ferramenta importante para monitorar e prevenir problemas críticos de saúde em diferentes perfis populacionais.

3.3 Exploração de Técnicas

O artigo de Dond (2025) explora o uso de técnicas de aprendizado de máquina para detectar hipertensão, aproveitando o poder dos algoritmos para analisar grandes conjuntos de dados, identificar padrões e fazer previsões com base em diversos fatores clínicos e de estilo de vida. Os dados dos pacientes examinados inclui medições de pressão arterial, informações demográficas e hábitos de vida.

O artigo de Gong, Chen e Ding (2023) procura desenvolver um método de predição de hipertensão mais confiável a partir de sinais não invasivos (Eletrocardiograma (ECG) e Foto Pletismograma (PPG)), utilizando inferência causal para selecionar variáveis realmente relacionadas à hipertensão, em vez de depender apenas da correlação.

[2] Disponível em: https://www.umass.edu/
[3] Disponível em: https://www.cdc.gov/nchs/nhanes/index.html

3.4 Contexto Quântico

Já no contexto de computação quântica, o trabalho de Gabrich, Freitas e Souza (2024) propõe uma abordagem que integra tecnologia quântica, algoritmos de aprendizado de máquina e *Clustered Regularly Interspaced Short Palindromic Repeats* (CRISPR), um sistema de edição genética. O objetivo é lidar com a manipulação precisa do *Deoxyribonucleic Acid* (DNA) para avanços na compreensão de distúrbios genéticos. Para isso, os autores desenvolvem uma rede neural clássica, uma rede internamente híbrida e outra rede internamente quântica para comparação de suas métricas.

3.5 Síntese da Literatura

A partir dos trabalhos de referência, é possível destacar que há vasta literatura que utiliza de aprendizado de máquina para classificar o risco de hipertensão em pacientes a partir de parâmetros selecionados que podem causar a presença da doença. Contudo, as contribuições dos trabalhos da literatura possuem resultados acima de 80%, no qual analisa de forma prioritária o uso de algoritmos de aprendizado profundo. O principal objetivo dos trabalhos é a detecção dos riscos de forma prematura em seus estágios iniciais.

No entanto, embora tenha um excesso de trabalhos na literatura, poucos exploram o uso da computação quântica em algoritmos de aprendizado de máquina. Este trabalho tem o intuito de explorar essa ferramenta utilizando conceitos da mecânica quântica para potencializar o desempenho do modelo na predição de riscos de hipertensão.

4 DESENVOLVIMENTO

Neste capítulo, na seção 4.1 discutimos os métodos sobre cada etapa deste trabalho. Na Seção 4.2 apresentamos a base de dados pública e utilizada no trabalho. Na Seção 4.3 é apresentada como foi construída a arquitetura da rede neural clássica. Na Seção 4.4 apresentamos a construção da rede neural híbrida, na qual uma das camadas ocultas utilizamos computação quântica. Por fim, na Seção 4.5 apresentamos como foi realizada a análise das *features* do conjunto de dados por meio da AI explicável.[4]

4.1 Metodologia

A metodologia deste trabalho seguiu as etapas práticas de um projeto completo de aprendizado de máquina, conforme descrito por Geron (2021, Capítulo 2), que são:

1. Visão geral do problema;
2. Obtenção dos dados;
3. Exploração e visualização dos dados;
4. Preparação dos dados para os algoritmos de ML;
5. Seleção e treinamento do modelo;
6. Ajuste fino (*fine-tuning*);
7. Apresentação da solução;
8. Implantação, monitoramento e manutenção do sistema;

Na etapa 1, buscou-se compreender o problema a ser resolvido e estabelecer a motivação científica e prática do estudo. O propósito do método é desenvolver modelos de aprendizado profundo: modelo clássico e modelo híbrido que integra recursos da computação quântica para detectar riscos de hipertensão em seus estágios iniciais.

Com isso, aplicamos técnicas de inteligência artificial para identificar padrões nos dados que possam antecipar a ocorrência da condição. A escolha por uma abordagem híbrida se justifica pela crescente da computação quântica na aplicação de modelos de aprendizado profundo. A partir da etapa 2 utilizamos um conjunto de dados de uma base pública, no qual aborda dados clínicos de pacientes. Os detalhes da base de dados apresentamos na seção 4.2.

Na etapa 3, de exploração dos dados, teve como objetivo compreender a distribuição das variáveis e identificar possíveis desbalanceamentos nas classes. O conjunto de dados foi

[4] As bibliotecas utilizadas e suas respectivas versões podem ser encontradas no Apêndice A.

carregado por meio da biblioteca *pandas* [5], escolhida, pois segundo Wes (2013) é amplamente reconhecida como uma das principais ferramentas de manipulação de dados em *Python*[6]. O *pandas* permite realizar operações vetorizadas, gerar estatísticas descritivas e produzir visualizações sobre diretamente os dados.

No contexto deste trabalho utilizamos o *pandas* para carregar o conjunto de dados de predição de hipertensão, explorar a distribuição das classes e identificar desbalanceamentos. Na etapa 4, foram aplicados transformações sobre o conjunto de dados disponíveis, visando a padronização e o tratamento dos dados a serem aplicados no treinamento do modelo.

Inicialmente, convertemos as variáveis definitivas em formato numérico por meio de codificação *one-hot*, e as variáveis-alvo "*Yes*" e "*No*" foram mapeadas para 1 e 0, respectivamente. Essa conversão é feita, pois os modelos operam sobre valores numéricos. Em seguida, os dados foram divididos em conjuntos de treino e teste utilizando a função *train_test_split* da biblioteca *scikit-learn*[7], para avaliar a capacidade de generalização do modelo. O conjunto de treino foi utilizado para o aprendizado dos padrões, enquanto o conjunto de teste permaneceu separado para validação. Essa separação reduz problemas como o sobreajuste.

Durante a preparação do conjunto de treino, observou-se um desbalanceamento entre as classes da variável alvo, com predominância de amostras de pacientes sem hipertensão. Para mitigar esse problema, foi aplicada a técnica *Synthetic Minority Over-sampling Technique* (SMOTE) [8], que gera amostras sintéticas da classe minoritária a partir da interpolação entre vizinhos mais próximos. Essa abordagem equilibra as classes e melhora a capacidade do modelo de identificar corretamente os casos positivos de hipertensão.

Por fim, os dados foram escalonados utilizando o método de padronização *StandardScaler*, também da biblioteca *scikit-learn*. Esse procedimento transforma cada variável para ter média zero e desvio padrão igual a um, mantendo a distribuição original dos dados e garantindo que todas as variáveis contribuam de forma equilibrada durante o processo de treinamento.

Na etapa 5, foi definida a arquitetura do modelo de aprendizado de máquina responsável por realizar as predições. Optou-se pela utilização de uma rede MLP. Definida por sua capacidade de modelar relações não lineares entre as variáveis e ajustar-se a diferentes padrões presentes nos dados. Implementamos os modelos utilizando as bibliotecas *TensorFlow* [9] e *Keras* [10],

[5] Disponível em: https://pandas.pydata.org/
[6] Disponível em: https://www.python.org/
[7] Disponível em: https://scikit-learn.org/stable/
[8] Disponível em: https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html
[9] Disponível em: https://www.tensorflow.org/
[10] Disponível em: https://keras.io/

amplamente empregadas em aplicações de aprendizado profundo. Essa escolha permitiu um controle adequado sobre as camadas da rede, funções de ativação e parâmetros de treinamento, além de oferecer suporte a aceleração via Unidade de Processamento Gráfico (GPU), que reduz o tempo de processamento.

Na etapa 6, os modelos passaram por ajustes dos seus hiper parâmetros e estratégias de regularização para reduzir problemas de sobreajuste, esses ajustes no número de camadas, neurônios e épocas foram realizados manualmente para encontrar o equilíbrio entre os dados da rede. Com isso, modelos com poucos parâmetros, ou seja, parâmetros muito simples tem o risco de não capturar padrões dos dados, enquanto modelos excessivamente complexos memorizam o conjunto de treino, reduzindo a capacidade de generalização.

Na etapa 7, após o treinamento e o ajuste do modelo, os resultados obtidos foram organizados e apresentados para sua interpretação. Buscamos nessa etapa validar o desempenho do modelo e demonstrar a efetividade das técnicas aplicadas. Foram utilizadas métricas de avaliação, como acurácia, precisão, sensibilidade e *F1-score*, essas métricas permitem uma análise detalhada do comportamento dos modelos tanto nos acertos quanto nos erros.

Na etapa 8, a última etapa, aplicamos a abordagem de Inteligência Artificial Explicável (XAI), para compreender e justificar as decisões tomadas pelo modelo. Essa fase torna-se necessária em aplicações na área da saúde, uma vez que não basta apenas obter previsões precisas, é preciso entender quais fatores influenciam o resultado, para manter transparência, confiança e suporte à tomada de decisões clínicas.

4.2 *Dataset* utilizado

A base de dados utilizada foi disponibilizada pelo site *Kaggle*[11], denominada: *Hypertension Risk Prediction Dataset*[12]. Ela contém 1.985 registros, 2 classes binárias e 11 variáveis geradas com base em percepções clínicas e padrões de dados de saúde pública, onde as variáveis do *dataset* são:

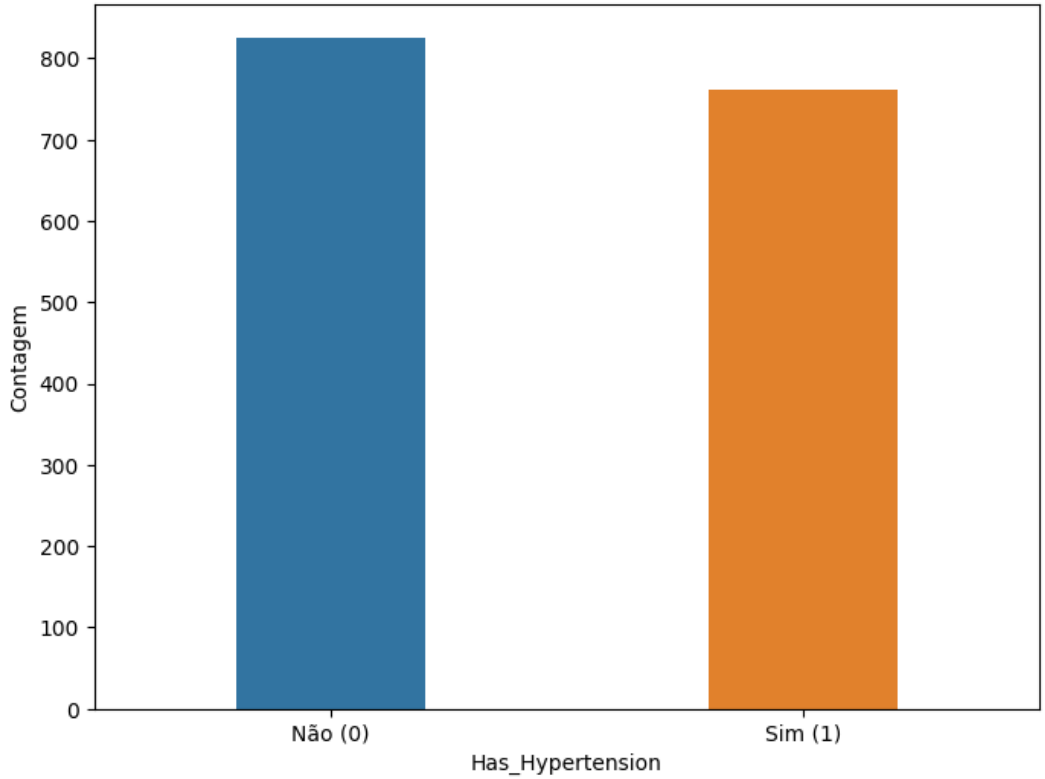
1. *Age*: Idade do paciente (em anos);
2. *Salt_Intake*: consumo diário de sal (em gramas);
3. *Stress_Score*: Escala de 0 a 10 que mede o nível de estresse psicológico;
4. *BP_History*: histórico de pressão arterial anterior: normal, pré-hipertensão, hipertensão;

[11] Disponível em: https://www.kaggle.com/
[12] Disponível em: https://www.kaggle.com/datasets/miadul/hypertension-risk-prediction-dataset

5. *Sleep_Duration*: Média de horas de sono por dia;
6. *BMI*: Índice de Massa Corporal (IMC), medida de obesidade baseada em peso/altura;
7. *Medication*: Tipo de medicação consumida (Nenhuma, Betabloqueador, Diurético, Inibidor da ECA, Outro);
8. *Family_History*: Histórico familiar de hipertensão;
9. *Exercise_Level*: Nível de atividade física (Baixo, Moderado, Alto);
10. *Smoking_Status* : Fumante ou Não Fumante;
11. *Has_Hypertension*: Presença de hipertensão;

O método utilizado foi dividir em duas classes, sendo a classe 0, para identificar que não há risco de hipertensão e a classe 1, em que há o risco de hipertensão. Também separamos os dados entre dados de treino e dados de teste, sendo 80% para treino e 20% para teste. O balanceamento entre as classes foi avaliado com o gráfico de distribuição, conforme ilustrado na Figura 2. Verificou-se um leve desequilíbrio entres as classes utilizadas para o treinamento, o que pode afetar na capacidade do modelo em generalizar para algumas categorias (Ali *et al.*, 2022).

Figura 2 – Distribuição das classes de treino.



Fonte: Autoria própria (2025).

4.3 Arquitetura da Rede Neural Clássica

Após o balanceamento dos dados, eles foram pré-processados com o uso da biblioteca *Scikit-Learn* do Python, utilizando o método *transform*, que retorna os dados de entrada modificados (Fabian, 2011). A construção do modelo ocorreu com uma camada de entrada, duas camadas ocultas, em que a primeira delas contém 64 neurônios, e a função de ativação *Rectified Linear Unit* (ReLU), empregada em redes neurais para aplicações não lineares, a segunda camada com 32 neurônios e também a ativação ReLU e por fim, uma camada de saída densamente conectada, com 1 neurônio, correspondendo ao número de saídas com a função de ativação *sigmoid*.

Para reduzir o *overfitting* nas camadas ocultas utilizamos o *dropout* [13] com probabilidade de 0,2. O otimizador usado foi o *adam* mantendo a taxa de aprendizado em 0.001. O treinamento deste modelo foi com *patience*=10, 100 épocas e *batch size* de 32.

Durante o treinamento, utilizamos a técnica de parada antecipada, que monitorou a perda de validação e restaurou os melhores pesos, interrompendo o treinamento. O desempenho sobre os dados de validação deixaram de melhorar, evitando o desperdício de recursos computacionais. Podemos observar a estrutura das camadas clássicas no Código-fonte 1.

Código-fonte 1 – Estrutura das camadas do modelo clássico.

```
model_classic = tf.keras.models.Sequential([
    tf.keras.layers.Input(shape=(input_dim,)),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(32, activation='relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(1, activation='sigmoid')
])

\\ Otimizador Adam
```

[13] Mais detalhes, vide Seção 2.2.1

4.4 Rede Neural Híbrida

A rede neural híbrida foi construída com o auxílio da *API* da *PennyLane* [14], pois oferece a capacidade de integrar camadas quânticas com redes neurais clássicas, no qual possibilita a implementação de redes híbridas quânticas. O método adotado para a tarefa de classificação de hipertensão com uma das camadas quânticas segue a arquitetura de classificadores quânticos centrados em circuito, conforme proposto por Schuld et. al. (Schuld *et al.*, 2020). Este paradigma utiliza um circuito quântico variacional como o núcleo de um modelo preditivo, integrado em um esquema de treinamento híbrido quântico-clássico.

O modelo foi executado em um hardware clássico, e por meio do *PennyLane* utilizamos o *default.qubit*, que é um simulador de computação quântica executado em computadores clássicos. A *wires=n_qubits* define o número de *qubit*s a serem utilizados, conforme ilustrado no Código-fonte 2.

Código-fonte 2 – Definição do ambiente de execução quântica e número de qubits.

```
1 import pennylane as qml
2 from pennylane.qnn import KerasLayer
3
4 n_qubits = 4
5 dev = qml.device("default.qubit", wires=n_qubits)
```

A estrutura do circuito quântico é definido por meio da função *qnode* (Código-fonte 3), uma função que indica o uso de conceitos quânticos. O circuito quântico é composto por uma camada de encaixe de ângulo (*AngleEmbedding*), em que esta camada converte dados clássicos em estados quânticos aplicando rotações em diferentes *qubits*. Em seguida, aplicamos uma camada de entrelaçamento forte (*StronglyEntanglingLayers*), para garantir que a informação não fique limitada em cada *qubit*.

Código-fonte 3 – Definição do circuito quântico.

```
1 @qml.qnode(dev, interface="tf")
2 def quantum_circuit(inputs, weights):
3     qml.AngleEmbedding(inputs, wires=range(n_qubits))
```

[14] Disponível em: https://pennylane.ai/

```
4       qml.StronglyEntanglingLayers(weights, wires=range(
            n_qubits))
5       return [qml.expval(qml.PauliZ(i)) for i in range(
            n_qubits)]
6
7   weight_shapes = {"weights": (1, n_qubits, 3)}
8   quantum_layer = KerasLayer(quantum_circuit, weight_shapes,
        output_dim=n_qubits)
```

No código-fonte 3, *qnode* um *decorador* que transforma a função `quantum_circuit` em um circuito quântico executável. Nos parâmetros do *decorador*, `dev` vincula o circuito ao simulador e a *interface* que recebe `'tf'` é informa ao *PennyLane* o uso do *TensorFlow*. Isso permite que o *TensorFlow* calcule os gradientes do circuito quântico automaticamente, tornando-o treinável em qualquer camada de uma rede neural.

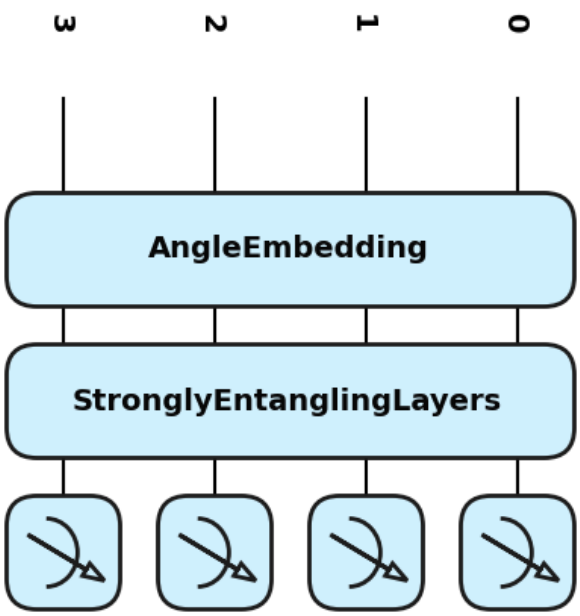
Em seguida, utilizamos a camada *AngleEmbedding* e a camada de processamento *StronglyEntanglingLayers*. Após essas camadas, extraímos um resultado clássico do circuito. A medição do valor extraído foi a partir do operador *Pauli Z* para cada *qubit*, conforme observado na Figura 3.

O modelo da rede neural híbrida, constitui-se 4 camadas, em que a primeira camada, é a camada de entrada, seguida por duas camadas ocultas, no qual a primeira é uma camada clássica densa, composta por 64 neurônios e com a função de ativação ReLU. Essa camada utiliza o *Batch Normalization*, um normalizador em lote que regulariza a saída dessa camada. Em seguida utiliza funções de regularização *Dropout* com probabilidade de 0,3 e o L2 para forçar a camada a aprender padrões mais simples e gerais.

A segunda camada oculta é a camada híbrida, que antecede o componente quântico emprega a função de ativação *Tangente Hiperbólica* (tanh), no qual a finalidade desta função é normalizar as saídas da camada para o intervalo de -1 a 1. Este é o formato para codificar informações em um circuito quântico.

Após a normalização, os valores de entrada são passados para a camada *AngleEmbedding*. A codificação dos dados é realizada mediante da porta de rotação *Pauli Z*, em que cada valor de entrada clássico determina o ângulo de rotação para um *qubit*. Esta técnica de codificação de fase altera a fase do estado quântico sem modificar as probabilidades de medição na base

Figura 3 – Estrutura do circuito com 4 *qubits*.



Fonte: Autoria própria (2025).

computacional. Essa representação é observada na Figura 4.

A porta *Pauli Z* também pode ser representada pelo circuito da Figura 5. Psi (ψ) representa o estado de um *qubit*, e seus respectivos valores são as amplitudes de probabilidade do estado do *qubit*. Por fim, temos a última camada, composta por um neurônio de saída, e a função de ativação: *sigmoid*. O otimizador utilizado foi o *Adam*, com taxa de aprendizado fixa em 0.001 para 100 épocas.

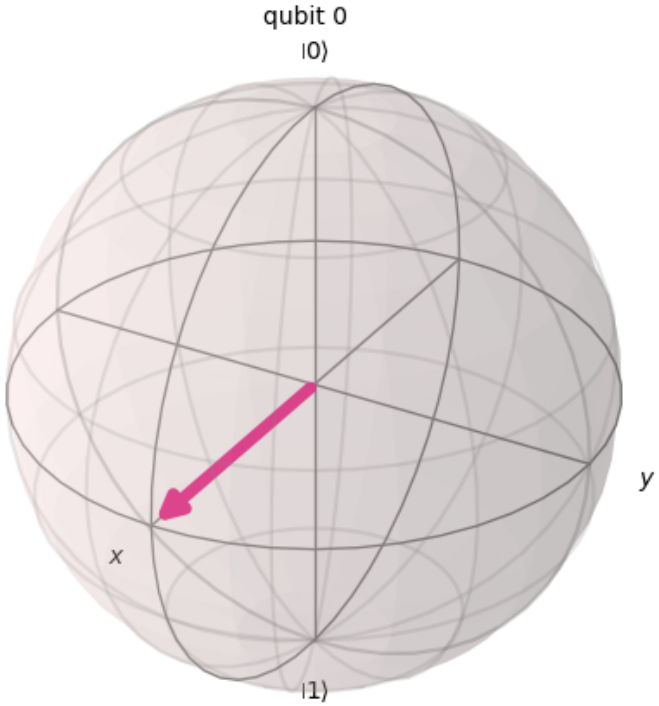
4.5 IA Explicável

Para interpretar o conjunto de dados, utilizamos o XAI. O método de XAI *SHapley Additive exPlanations* (SHAP), é um método cujo objetivo é explicar a previsão de uma instância calculando a contribuição de cada recurso para previsões individuais e de comportamento geral do modelo (Lundberg; Lee, 2017). Uma previsão pelo SHAP é expressa matematicamente pela Equação 4.1.

$$\phi_j(x) = \sum_{S \subseteq N \setminus \{j\}} \frac{|S|!\,(|N| - |S| - 1)!}{|N|!} \left[f(S \cup \{j\}) - f(S) \right] \tag{4.1}$$

Na Equação 4.1, $\phi_j(x)$ indica o valor de SHAP associado à característica j. O conjunto N

Figura 4 – Representação do estado de um *qubit* na Esfera de *Bloch* após aplicação da porta *Pauli Z* - o vértice aponta para o eixo X negativo, ilustrando a rotação de 180° no plano XY causada pela porta *Pauli Z* no estado de superposição.



Fonte: Autoria própria (2025).

Figura 5 – Circuito quântico demonstrando estado de um *qubit* ligado a porta *Pauli Z*.

Fonte: Autoria própria (2025).

corresponde ao total de atributos considerados no modelo, enquanto S representa um subconjunto de características que não contém j. A função $f(S)$ expressa a previsão obtida pelo modelo ao utilizar apenas as variáveis pertencentes a S, e o termo $f(S \cup j) - f(S)$ mede a contribuição marginal da característica j quando ela é incorporada a esse subconjunto.

Na aplicação do SHAP o modelo treinado é carregado por meio da função *load_model* do *tf keras*. Em seguida, os dados de treino e teste são lidos para que os nomes das colunas ajustadas tenha consistência entre os conjuntos. Na etapa seguinte, selecionamos um subconjunto de dados chamado: *background data* utilizando `shap.sample(input_test, 100)`. Esse subconjunto é usado pelo SHAP para calcular as contribuições de cada variável.

Em seguida, aplicamos o método genérico *KernelExplainer* para estimar os valores de

SHAP por meio de amostragens e aproximações da contribuição de cada variável em relação à previsão final. Esses valores de SHAP são calculados como demonstra o código-fonte 4.

Código-fonte 4 – Inicialização do *KernelExplainer* do SHAP.

```
1 explainer = shap.KernelExplainer(model.predict,
    background_data)
```

5 RESULTADOS E DISCUSSÕES

Neste capítulo, apresentamos os dados obtidos e a análise dos testes realizados após a construção dos modelos de DL descritos nas Seções 4.3 e 4.4 do Capítulo 4. O objetivo dos testes de avaliação dos modelos foi identificar quais usuários, da base de dados, detinham risco de hipertensão a partir das classes de influência, e detectar quais parâmetros exerciam maior impacto. Para isso realizamos as medidas da matriz de *confusão*, *acurácia*, *precisão*, *sensibilidade* e *F1-score* para cada modelo.

Essas métricas permitem avaliar o desempenho de um modelo de classificação, considerando que, em uma classificação binária, existe uma classe positiva e uma classe negativa, sendo o objetivo principal identificar corretamente a classe positiva. Por esse motivo, é necessário analisar o número total de acertos (acurácia). Contudo, avaliar apenas a acurácia é insuficiente em situações de desequilíbrio entre as classes, pois o modelo pode apresentar alta acurácia mesmo classificando incorretamente a classe menos frequente.

A *sensibilidade* indica a capacidade do modelo de identificar corretamente os indivíduos que possuem risco de hipertensão. A *precisão* representa a proporção de previsões positivas que estão corretas, ou seja, quantos dos indivíduos classificados como de risco realmente pertencem a essa categoria. O *F1-score* combina sensibilidade e precisão em uma medida única, permitindo avaliar o desempenho do modelo em cenários com classes desbalanceadas.

Por fim, este capítulo está organizado na seguinte maneira: na Seção 5.1 descrevemos os resultados e as métricas de desempenho utilizadas na avaliação do modelo clássico. Na Seção 5.2 apresentamos os resultados dos testes com o modelo híbrido. Ademais, na Seção 5.3, mostramos a comparação de desempenho entre os modelos. Por fim, na Seção 5.4, realizamos uma análise da influência de cada variável da base de dados com a AI explicável

5.1 Testes com o modelo clássico

O objetivo dos testes com o modelo clássico foi avaliar os resultados obtidos pelas métricas. Esses resultados foram adquiridos por meio da função *classification_report* (da biblioteca *scikit-learn*), durante o treinamento do modelo, essa função fornece o resultado para cada métrica. A partir da Tabela 4, pode-se inferir as estatísticas obtidas para as medidas.

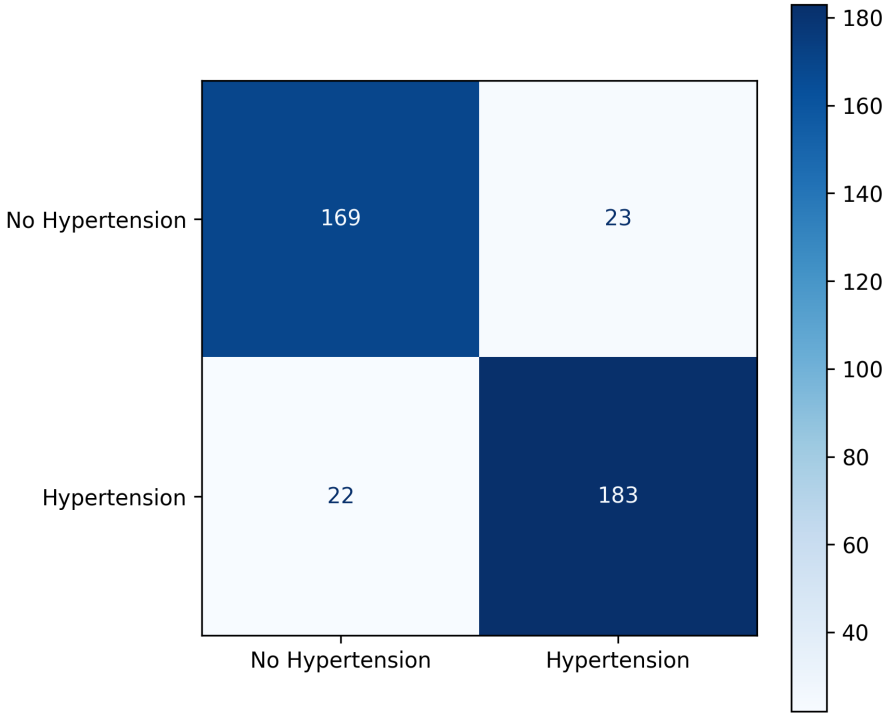
A partir da Tabela 4, foi possível calcular tanto a média simples quanto a média ponderada das métricas. Na média simples, cada classe possui o mesmo peso, independentemente da

Acurácia (%)	Sensibilidade (%)	Precisão (%)	F1-score (%)
89%	88%	90%	89%

Tabela 4 – Métricas de desempenho do modelo clássico obtidas após aplicação da metodologia do capítulo 4 sobre a base de dados da Seção 4.2.

quantidade de amostras, resultando em um valor final de 89%. Já na média ponderada, a classe com maior número de amostras exerce maior influência no cálculo, obtendo-se também 89% de desempenho. Essas métricas derivam-se da matriz de confusão, em que contém o número de acertos e erros do modelo para cada classe, conforme ilustrado na Figura 6.

Figura 6 – Matriz de confusão para modelo clássico.

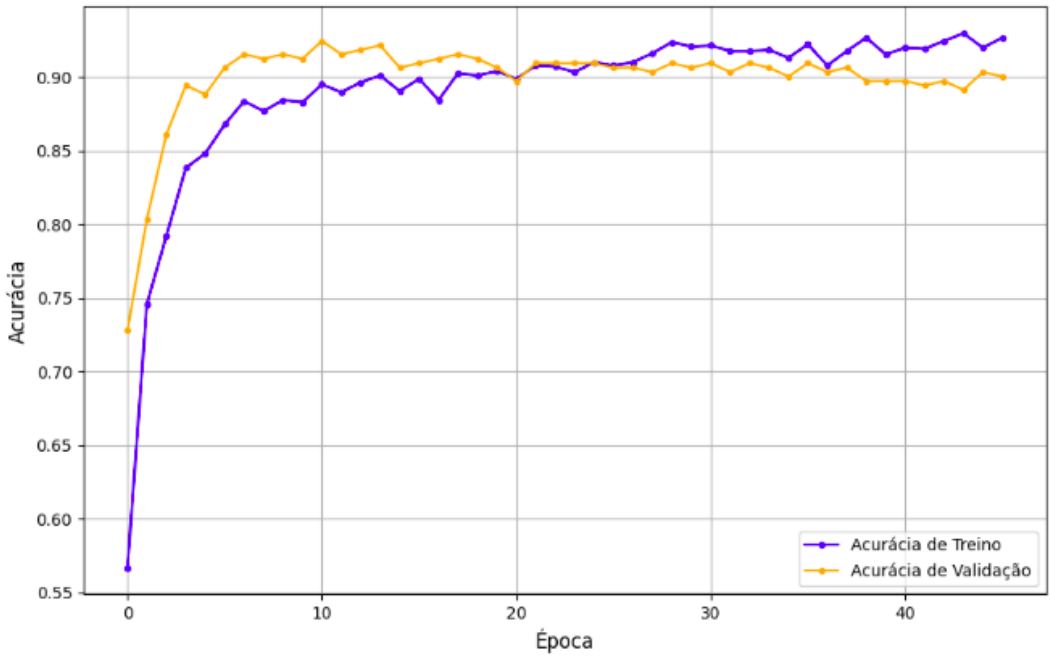


Fonte: Autoria própria (2025).

Na Figura 6, podemos observar os acertos e erros do modelo clássico, para as variáveis de *Hypertension* e *No Hypertension*, nos quais os valores na diagonal principal correspondem aos verdadeiros positivos e verdadeiros negativos, ou seja, os valores que o modelo acertou. Já na diagonal secundária referem-se aos falsos positivos e falsos negativos, valores que o modelo errou. Além da matriz de confusão, a Figura 7 apresenta a variação da acurácia dos dados de treino e de teste (validação), no decorrer do número de épocas.

Na Figura 7, observa-se que a acurácia do conjunto de treino cresce rapidamente durante os primeiros instantes. Inicialmente, a acurácia parte de cerca de 0,55 (55%) e estabiliza-se em torno de 0,90 (90%) após aproximadamente 10 épocas. A acurácia de validação segue um

Figura 7 – Acurácia dos dados do modelo clássico por época.



Fonte: Autoria própria (2025).

comportamento semelhante ao de treino, em que alcança valores próximos de 0,90, o que indica que o modelo conseguiu aprender os padrões reais e não apenas memorizou os exemplos do treino.

Para complementar a análise da acurácia, a Figura 8 apresenta a evolução da função de perda ao longo das épocas. Observa-se uma redução progressiva dos valores de perda ao decorrer das épocas tanto para os dados de treinamento quanto para validação, indicando que o modelo conseguiu minimizar o erro de predição durante o aprendizado.

5.2 Testes com o modelo híbrido

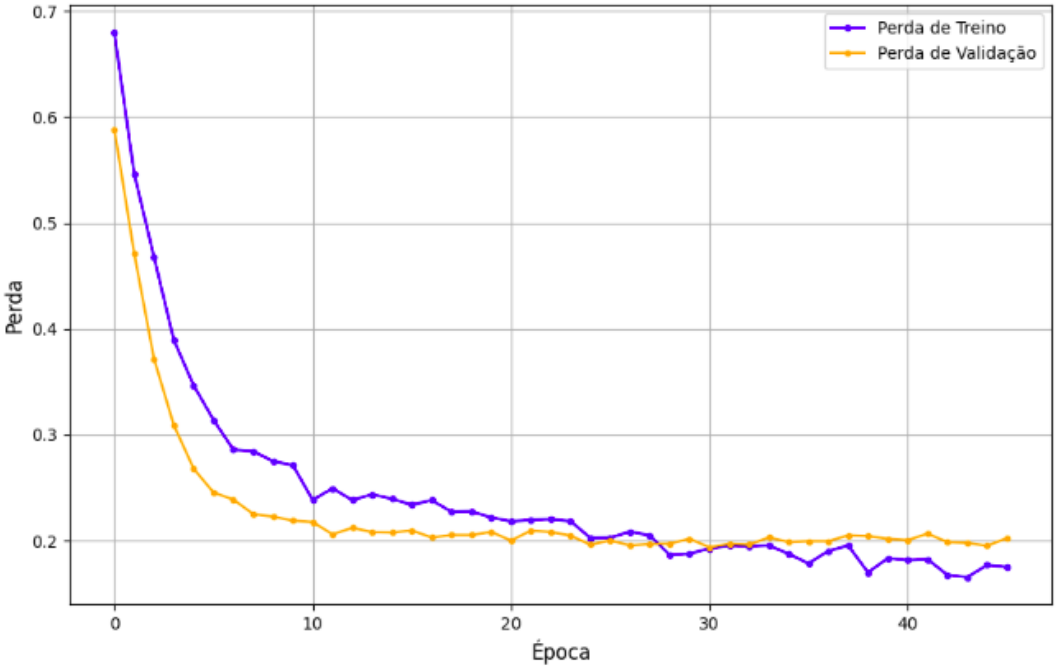
Os testes com o modelo híbrido incluem a análise dos valores das métricas obtidas (acurácia, sensibilidade, precisão e *F1-score*). Os dados apresentados na Tabela 5, indicam o percentual de cada métrica.

Acurácia (%)	Sensibilidade (%)	Precisão (%)	F1-score (%)
90%	90%	90%	90%

Tabela 5 – Métricas de desempenho do modelo híbrido obtidas após aplicação da metodologia do capítulo 4 sobre a base de dados da Seção 4.2.

A partir da Tabela 5, pode-se inferir um equilíbrio entre a média das medidas. Na Figura 9 podemos observar os valores precisos das medidas para cada uma das duas classes, em que o

Figura 8 – Função perda do modelo clássico por época.



Fonte: Autoria própria (2025).

modelo mantém resultados próximos para ambas.

Figura 9 – Classificação *Report*.

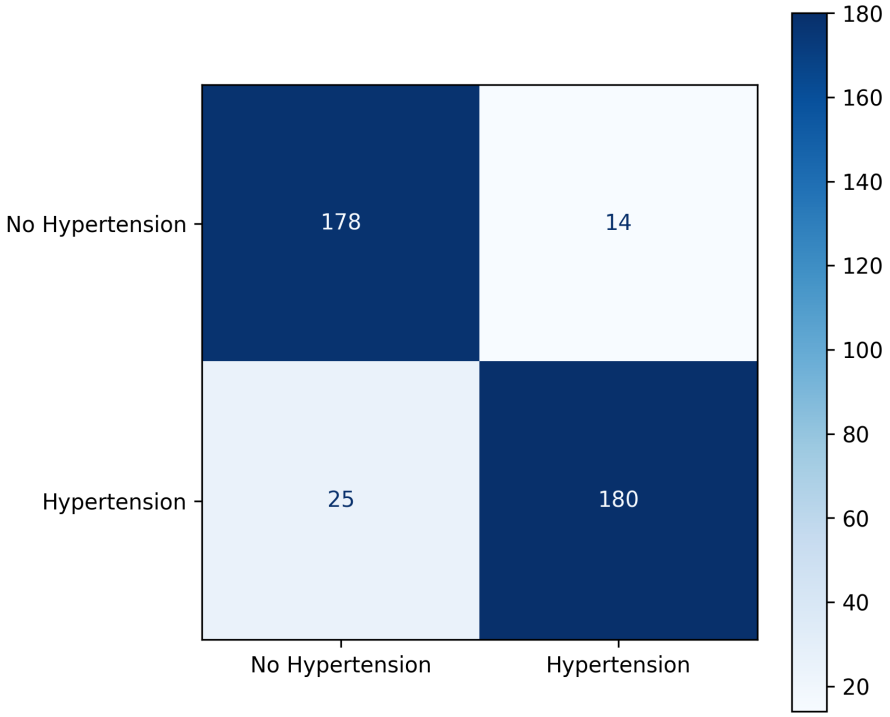
	precision	recall	f1-score	support
No Hypertension	0.88	0.93	0.90	192
Hypertension	0.93	0.88	0.90	205
accuracy			0.90	397
macro avg	0.90	0.90	0.90	397
weighted avg	0.90	0.90	0.90	397

Fonte: Autoria própria (2025).

Conforme ilustrado na Figura 9, observa-se que o modelo obteve uma precisão de 93% para os casos em que se tem hipertensão. No entanto, para casos em que não se tem hipertensão, foi de 88%. Para a sensibilidade, obtivemos o contrário da precisão, com 93% para não hipertensão e 88% para hipertensão, as demais medidas se mantiveram com valores em 90%. A matriz de confusão é um classificador dos acertos e erros para cada classe na Figura 10.

A partir da Figura 10, pode-se constatar que houve um número maior de acertos na diagonal principal, referente aos verdadeiros positivos e verdadeiros negativos, e na diagonal secundária, obteve-se um número menor para os falsos positivos e falsos negativos. Outra alternativa para avaliação das métricas é o gráfico da acurácia por época, conforme ilustrado na

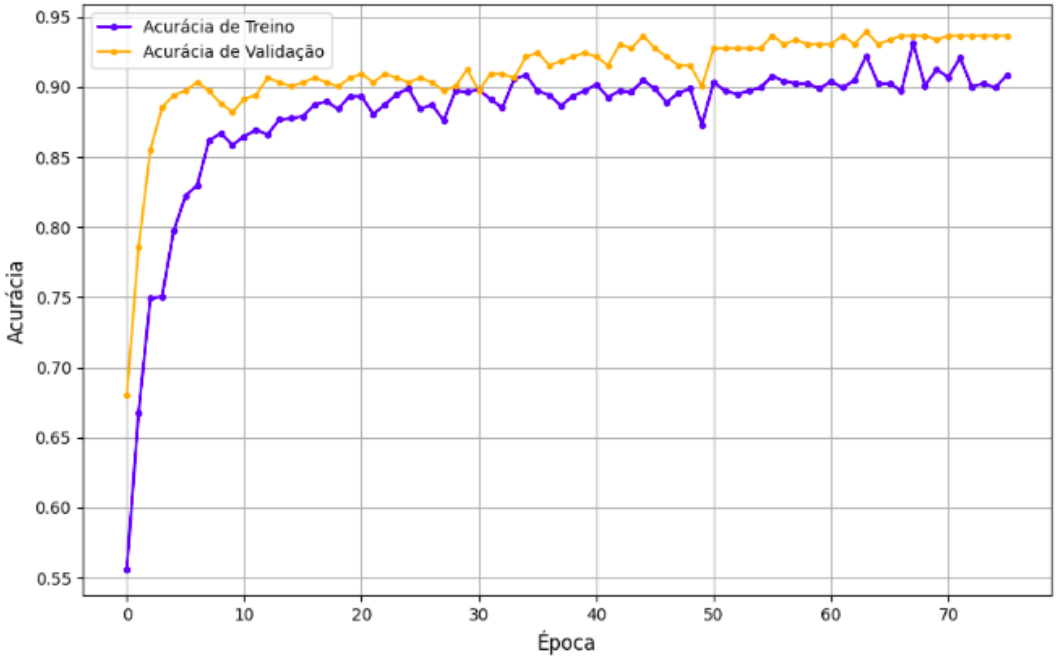
Figura 10 – Matriz de confusão para modelo híbrido.



Fonte: Autoria própria (2025).

Figura 11.

Figura 11 – Acurácia dos dados para o modelo híbrido por época.



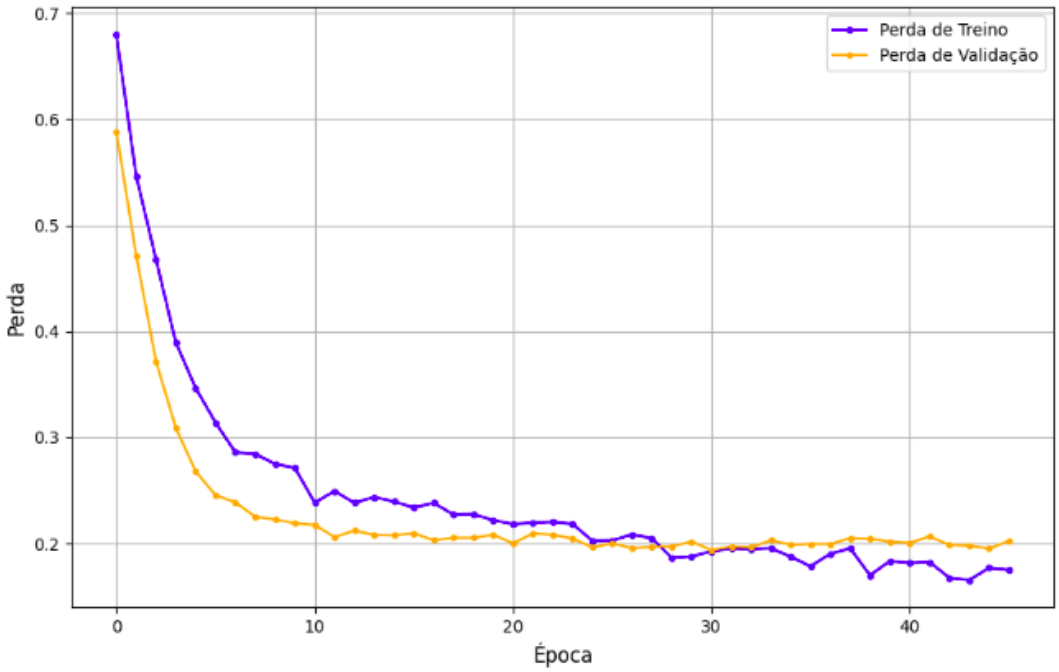
Fonte: Autoria própria (2025).

Na Figura 11, identifica-se que a acurácia dos dados de treino cresce rapidamente dentro das primeiras 10 épocas. Pode-se observar uma estabilidade entre a época 20 e a época 60, uma

acurácia por volta de 0,90 (90%). A acurácia de validação conseguiu atingir sua estabilidade por volta dos 0,90 antes da época 10, e se manteve acima dos 90% a partir da época 30. Também nota-se que o mecanismo de parada antecipada ocorreu entre 70 e 80 épocas.

Além da acurácia, foi medida a função perda por épocas, conforme ilustrado na Figura 12. Verifica-se uma redução da perda, iniciando em 0,7 e se mantendo de forma constante quando atinge 0,2 entre a época 20 e 30 para os dados de treinamento, e por volta da época 10 para os dados de validação.

Figura 12 – Função perda para o modelo híbrido por época.



Fonte: Autoria própria (2025).

5.3 Comparação entre os modelos

Ao analisar os resultados obtidos para os modelos clássicos e híbridos, possibilitou-se realizar uma comparação entre ambos os modelos. Confeccionamos a Tabela 6 para comparação do tempo de execução final, que permite avaliar a eficiência dos modelos ao longo das épocas, considerando instantes iniciais, intermediários e finais do treinamento, bem como o custo de processamento necessário para atingir o desempenho final.

A partir da Tabela 6, pode-se inferir que o modelo híbrido, embora tenha apresentado um desempenho inicial inferior ao do modelo clássico em termos de acurácia e função de perda, mostrou melhorias ao longo das épocas. No entanto, o tempo de processamento foi

Tabela 6 – Comparação Detalhada de Desempenho entre os Modelos.

	Clássica					Híbrida				
Métrica	**Época(1)**	**Época(20)**	**Pico (Perda)**	**Pico (Acur.)**	**Valor final**	**Época(1)**	**Época(20)**	**Pico (Perda)**	**Pico (Acur.)**	**Valor final**
Perda	0.6130	0.2066	0.1890	0.1900	0.2381	0.8581	0.2867	0.2383	0.2470	2660
Acurácia	0.7583	0.9124	0.9124	0.9215	0.8866	0.6103	0.9154	0.9215	0.9305	0.9018
F1-Score	...	...	...	...	0.8905	...	...	...	...	0.9023
Tempo	...	3.52s	5.66s	7.05s	7.68s	...	6m 41s	19m 41s	16m 1s	22m 2s

significativamente maior, cerca de aproximadamente 172 vezes superior ao do modelo clássico.

Esse fator é atribuído à complexidade adicional da camada de processamento quântico e da simulação envolvida. Considera-se, de forma hipotética, que ao empregar um hardware quântico físico, o tempo de execução poderia se tornar comparável ao de um modelo clássico, uma vez que os *qubits* seriam mapeados diretamente nos componentes físicos do processador. No entanto, essa hipótese não foi verificada no presente trabalho, representando uma limitação que poderá ser explorada em investigações futuras.

Portanto, de forma geral, o modelo de rede neural híbrida apresentou uma acurácia final levemente superior à rede clássica. Embora o treinamento de modelos híbridos em hardwares clássicos seja um desafio, aponta para o potencial das técnicas quânticas em melhorar o desempenho preditivo.

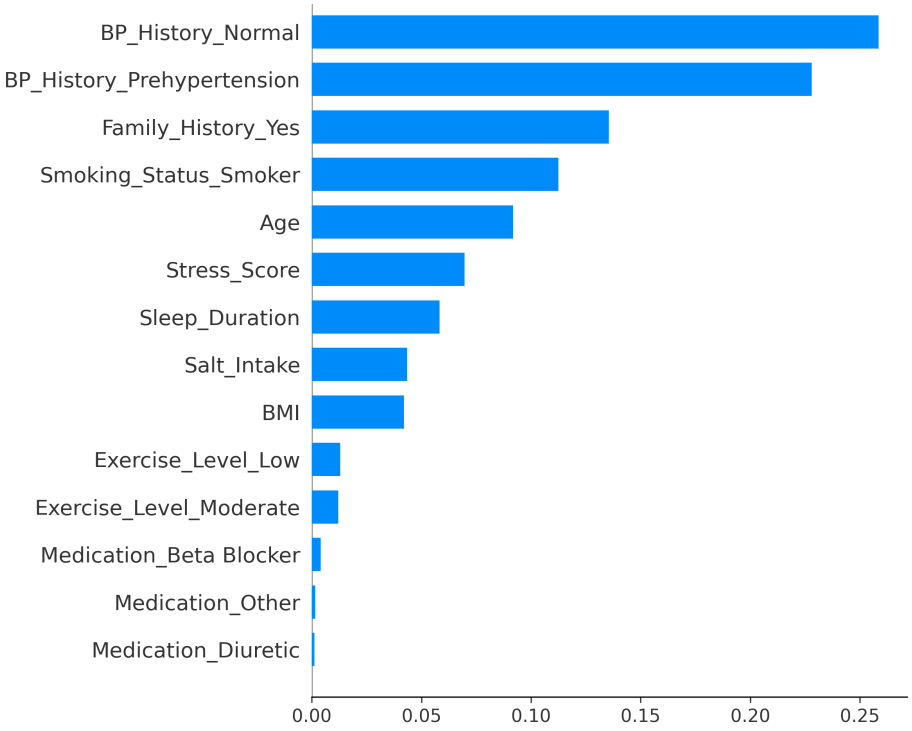
5.4 Testes com a IA explicável

A partir dos testes realizados com a abordagem da IA explicável, utilizando o método SHAP para avaliar a influência de cada variável nas predições. Identificamos os atributos de entrada que causam maior e menor impacto no diagnóstico da hipertensão, e está descrito na Figura 13. As variáveis relacionadas ao histórico de pressão normal, de pré-hipertensão e histórico familiar positivo são as que mais influenciam para o modelo prever o risco de hipertensão.

Conforme ilustrado na Figura 13, observa-se que após o alto impacto que o histórico familiar tem, os próximos fatores de maior influência é o uso de cigarro, a idade e o nível de estresse.

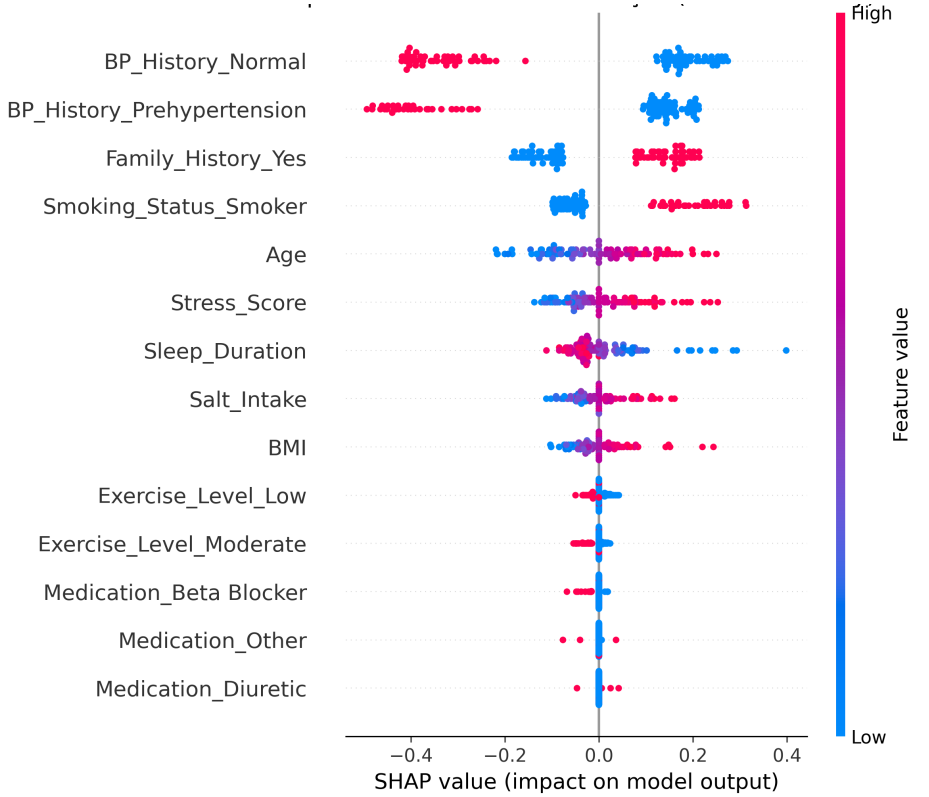
A Figura 14 ilustra de forma mais detalhada como os valores de cada variável impacta na previsão da hipertensão. Nela, os valores mais alto (em vermelho) dessas variáveis estão fortemente associados a um aumento na probabilidade de diagnóstico de hipertensão. Já os valores mais baixos (em azul) têm impacto negativo, ou seja, diminuem a probabilidade do diagnóstico.

Figura 13 – Variáveis que mais influenciam no diagnóstico positivo e negativo para riscos de hipertensão.



Fonte: Autoria própria (2025).

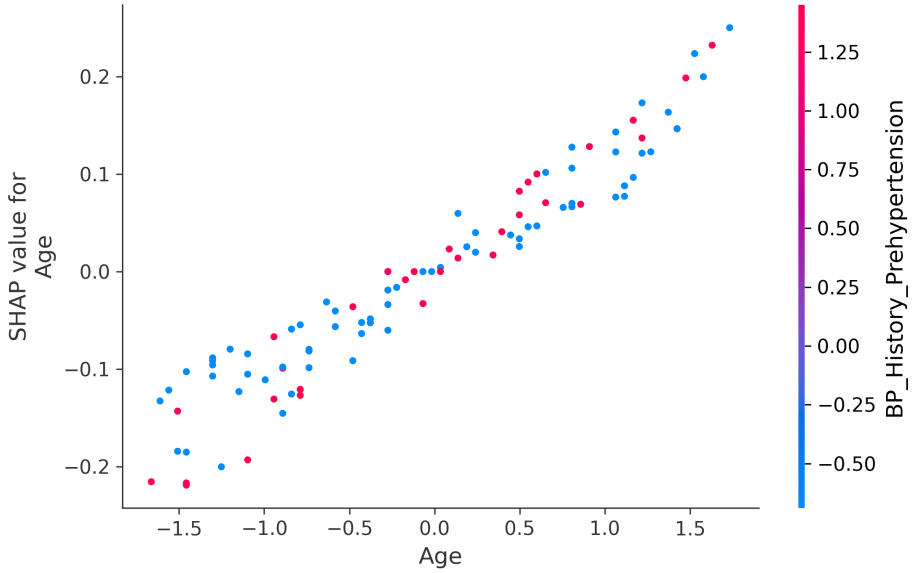
Figura 14 – Variáveis que mais impactam no diagnóstico de hipertensão.



Fonte: Autoria própria (2025).

Para analisar como o modelo utiliza cada variável, utilizamos a ferramenta do SHAP para gerar gráficos de dependência. Na Figura 15, observa-se que o modelo identificou que embora a idade represente um fator de risco, a presença de histórico de pré-hipertensão reduz levemente o impacto desse risco. No entanto, por esse resultado parecer contraintuitivo, é justificado pelo fato que a ausência de pré-hipertensão corresponda a pacientes com hipertensão já estabelecida.

Figura 15 – Dependência SHAP: *Age*.



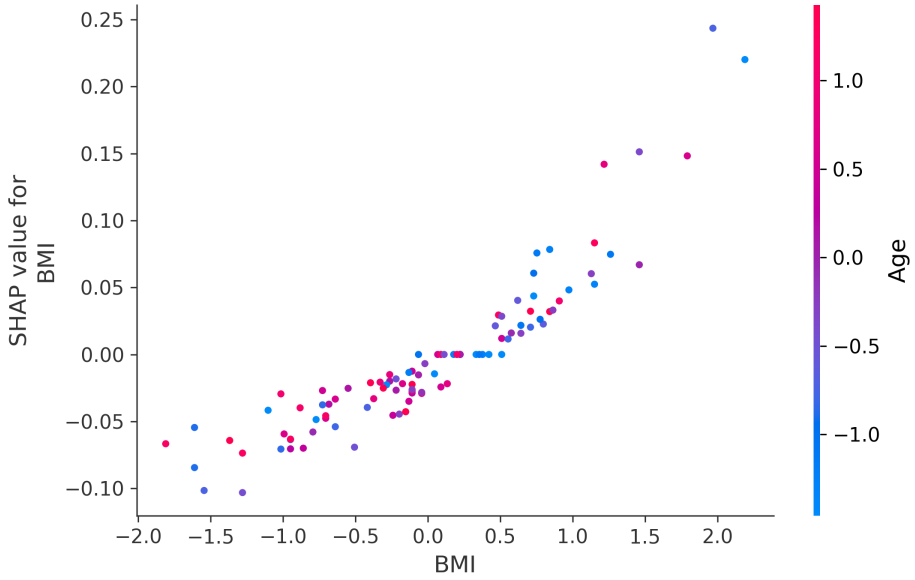
Fonte: Autoria própria (2025).

Na Figura 16, infere-se que conforme o *BMI* aumenta no eixo x, o valor do SHAP para *BMI* também aumenta no eixo y. O modelo aprendeu que quanto maior for o *BMI*, maior será a probabilidade de hipertensão positiva. Também observamos que o modelo não apenas aprendeu que um *BMI* alto é ruim, mas que é principalmente ruim para pessoas com a idade mais avançada.

A Figura 17 apresenta um gráfico com características de uma variável binária. Quando *BP History Normal* é 1, ou seja, está à direita, com o valor positivo no eixo x, os valores SHAP são negativos. Isso significa que ter um histórico normal é um fator de proteção contra o risco de hipertensão. A análise de interação com *Salt Intake* indica que o impacto do histórico de pressão é dominante sobre o consumo de sal, ou seja, ter um histórico normal ameniza o risco de hipertensão perante o consumo de sal.

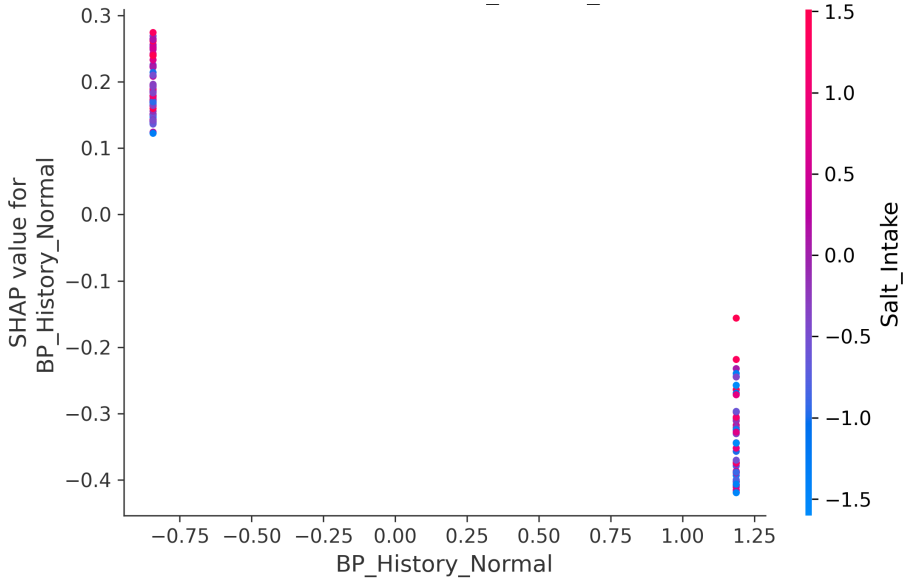
A Figura 18 mostra que a principal *feature* é *BP History Prehypertension*, na qual, quando é 1, os valores SHAP são negativos. O modelo considera ter tido pré-hipertensão como um fator que reduz o risco em comparação com outras situações, no caso da hipertensão já existente. Na *feature* de interação: *BP History Normal*, o grupo da direita com pré-hipertensão, todos os pontos são azuis, pois é impossível ter um histórico normal e de pré-hipertensão ao mesmo tempo.

Figura 16 – Dependência SHAP: *BMI*.



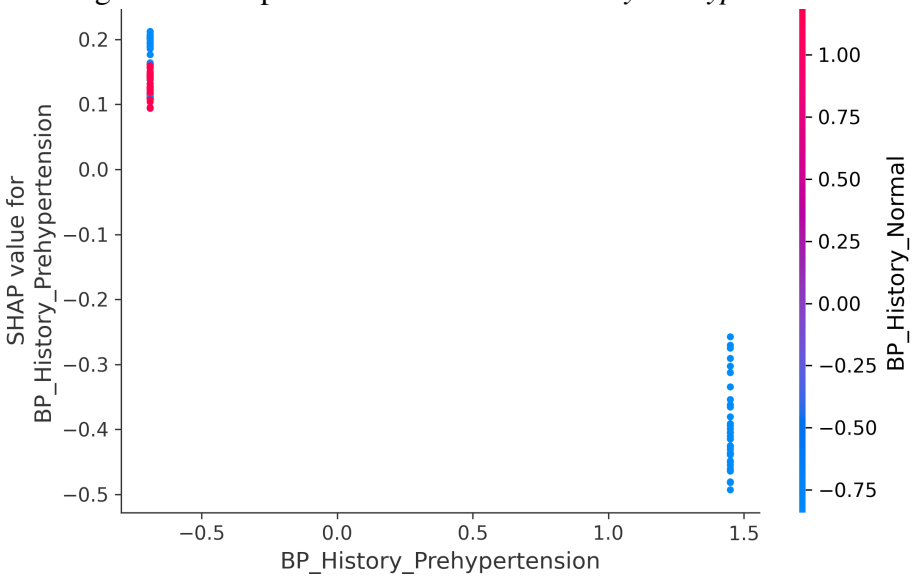
Fonte: Autoria própria (2025).

Figura 17 – Dependência SHAP: *BP_History Normal*.



Fonte: Autoria própria (2025).

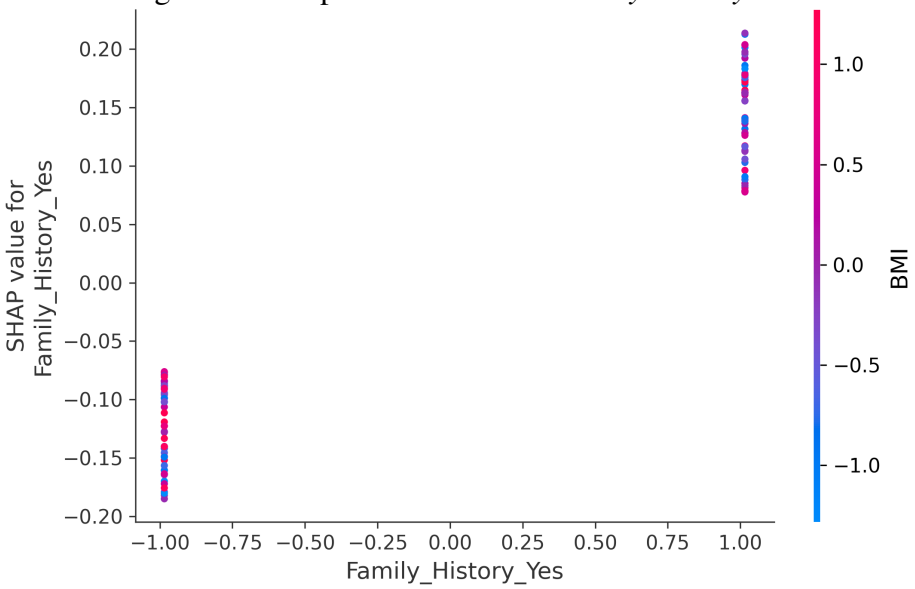
Figura 18 – Dependência SHAP: *BP History Prehypertension*.



Fonte: Autoria própria (2025).

A Figura 19 tem a *feature: Family History Yes*, na qual, quando é 1 à direita com valor positivo, os valores SHAP são positivos. Isso quer dizer que ter histórico familiar aumenta o risco previsto. Quando o valor é 0, ou seja, a esquerda com valor negativo, os valores SHAP são negativos. Isso implica que não ter um histórico familiar com hipertensão é um fator de proteção. A interação com o *BMI* sugere que o modelo atribui um risco maior a pacientes com histórico familiar que também possuem um *BMI* elevado.

Figura 19 – Dependência SHAP: *Family History Yes*.

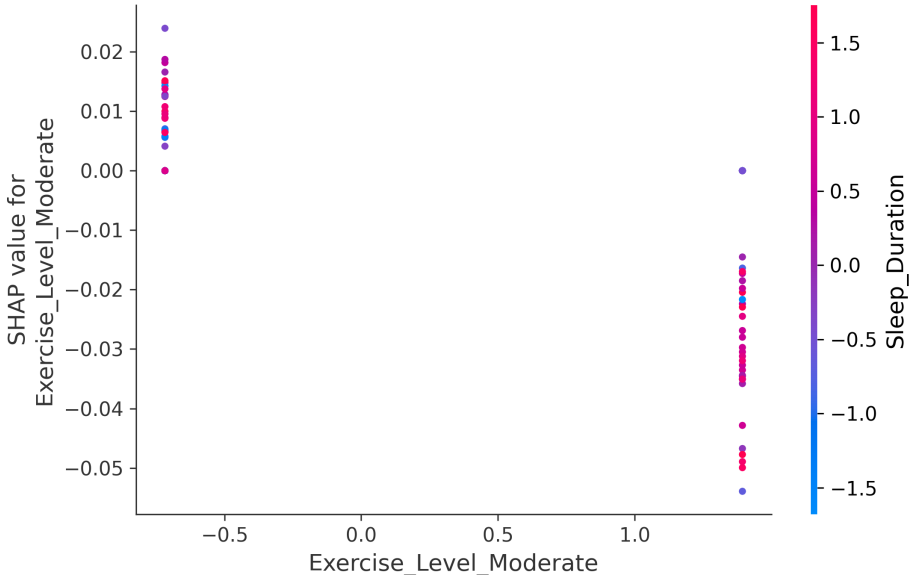


Fonte: Autoria própria (2025).

A Figura 20 tem como *feature* principal: *Exercise Level Moderate*. Quando a *feature*

é 1, à direita, os valores SHAP estão em torno de zero. Isso implica que ter nível de exercício moderado tem um impacto neutro. Como os valores permanecem próximos a zero e nenhuma interação significativa com a *feature Sleep Duration* indica uma contribuição de baixo impacto para as predições.

Figura 20 – Dependência SHAP: *Exercise level moderate*.



Fonte: Autoria própria (2025).

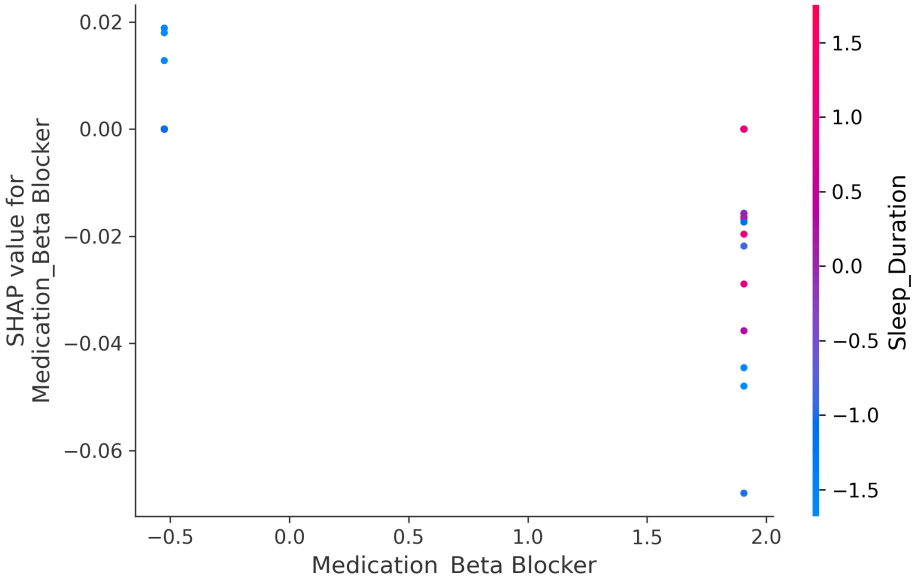
A Figura 21 apresenta como *feature* principal *Medication Beta Blocker*. Essa variável demonstrou ter pouca influência nas predições. O gráfico de dependência mostra valores SHAP aproximadamente em torno do zero, sem relação direta com o valor da *feature* ou com a *Sleep Duration*.

Na Figura 22, observa-se que a *feature* principal é *Smoking Status Smoker*. Essa variável revela que, quando o valor é 1, à direita, indicando efeito positivo, os valores de SHAP aumentam, o que sugere que ser fumante eleva o risco de hipertensão. Já a *feature* de interação *BP History Normal* evidencia que o tabagismo é ainda mais prejudicial em indivíduos sem histórico de pressão normal.

A Figura 23, tem como *feature* principal: *Stress Score*. Conforme o nível da variável no eixo x aumenta, o valor SHAP no eixo y tende a aumentar também. Com isso, o nível de estresse mais alto contribui para um maior risco previsto. A análise de interação com *BP History Prehypertension* não revelou um padrão distinto. Isso indica que o impacto do estresse é largamente independente do histórico de pré-hipertensão.

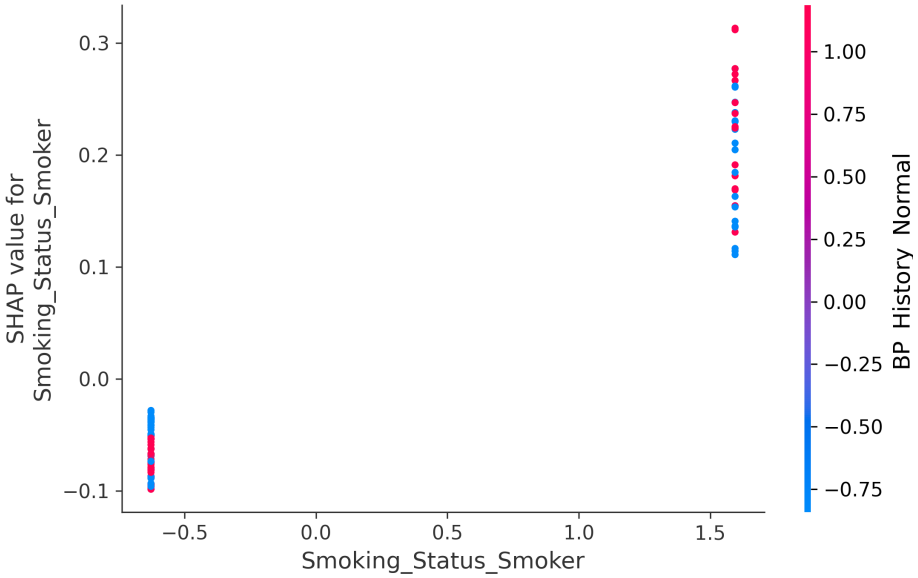
A Figura 24, tem como *feature* principal: *Sleep Duration*. A partir da análise realizada,

Figura 21 – Dependência SHAP: *Medication Beta Blocker*.



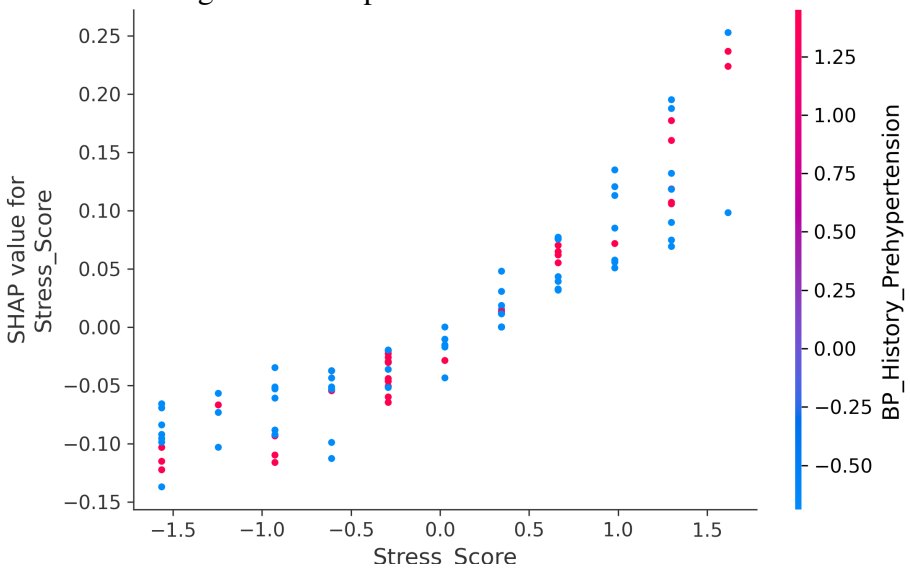
Fonte: Autoria própria (2025).

Figura 22 – Dependência SHAP: *Smoking Status Smoker*.



Fonte: Autoria própria (2025).

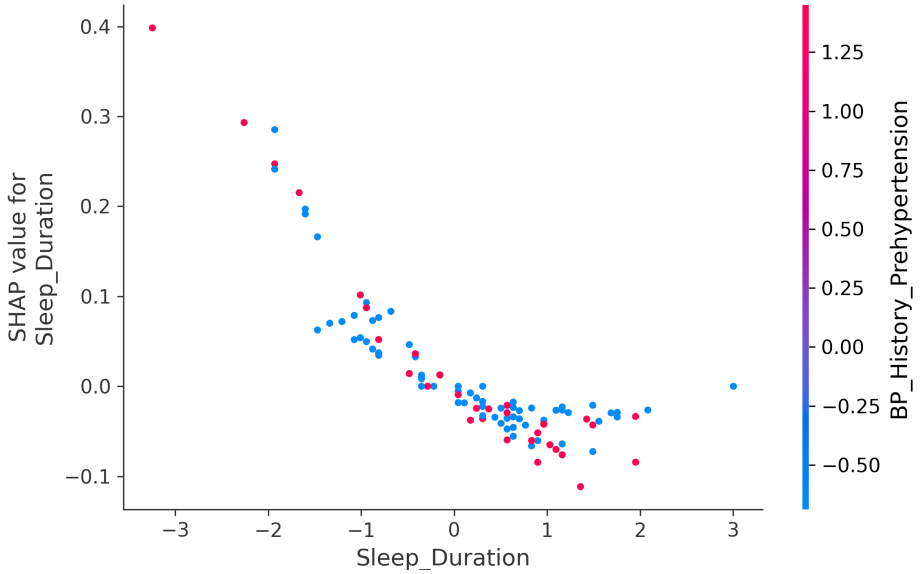
Figura 23 – Dependência SHAP: *Stress Score*.



Fonte: Autoria própria (2025).

podemos observar que para durações de sonos muito baixos, ou seja, valores negativos no eixo *x*, o impacto no risco é alto. Dormir muito pouco aumenta o risco. As interações com *BP History Prehypertension* não trazem impacto.

Figura 24 – Dependência SHAP: *Sleep Duration*.

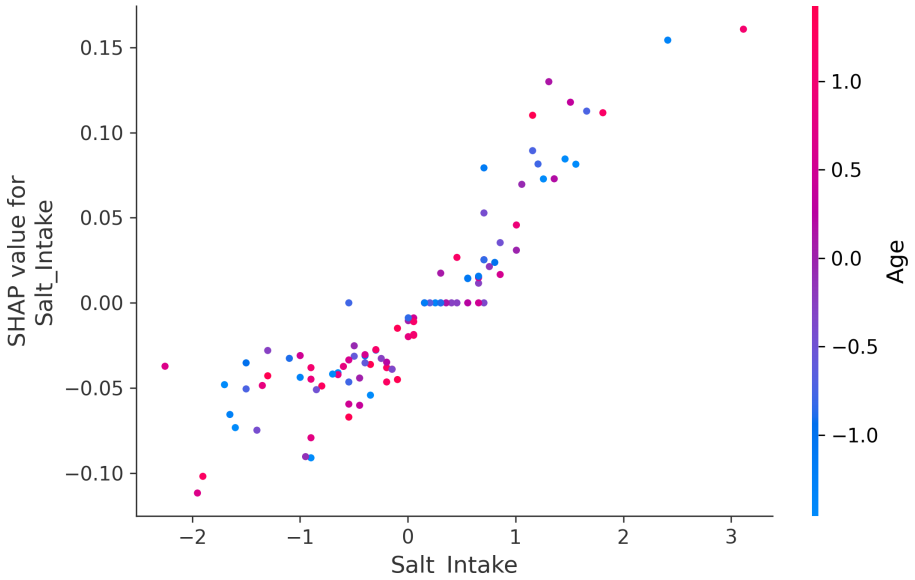


Fonte: Autoria própria (2025).

A Figura 25, tem como principal *feature*: *Salt Intake*, no qual conforme o consumo de sal no eixo *x* aumenta, o valor SHAP no eixo *y* também tende a aumentar. Maior consumo de sal contribui para um maior risco previsto. A variável *Age* tem impacto do consumo elevado de sal, em que influência em pacientes de idade mais elevada.

As variáveis restantes: *Exercise Level Low, Medication Diuretic e Medication Other*, têm

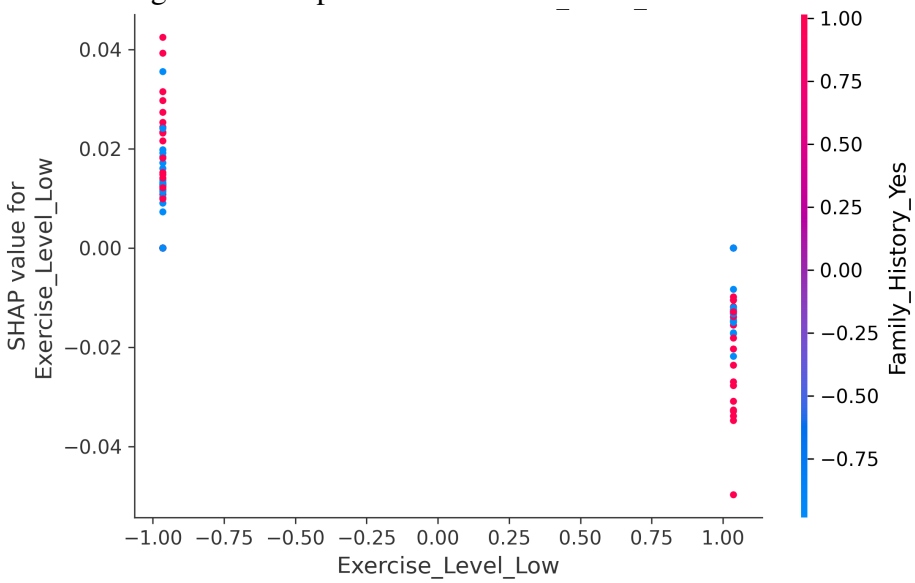
Figura 25 – Dependência SHAP: *Salt Intake*.



Fonte: Autoria própria (2025).

pouca influência no modelo, assim como suas *features* de interação. Nas Figuras 26, 27, 28, podemos observá-las no gráfico de dependências.

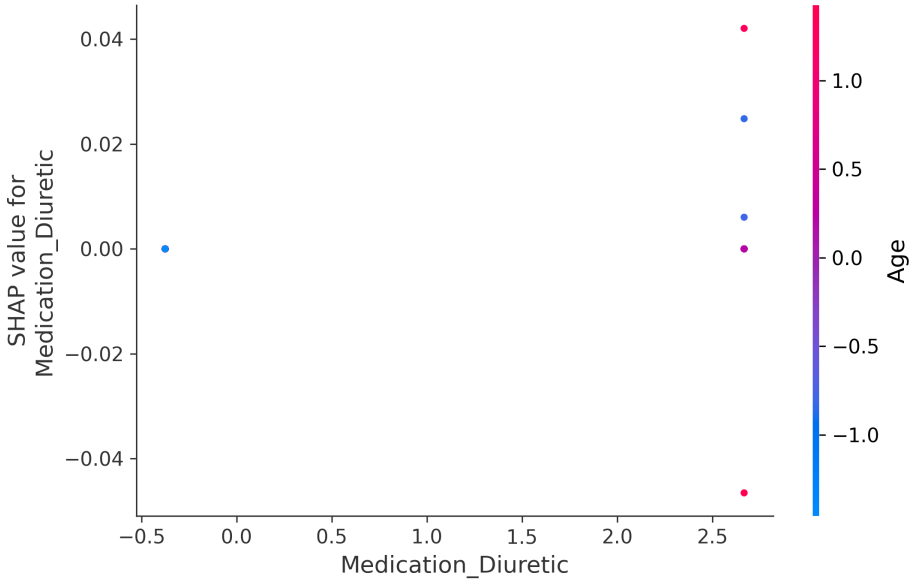
Figura 26 – Dependência SHAP: *Exercise Level Low*.



Fonte: Autoria própria (2025).

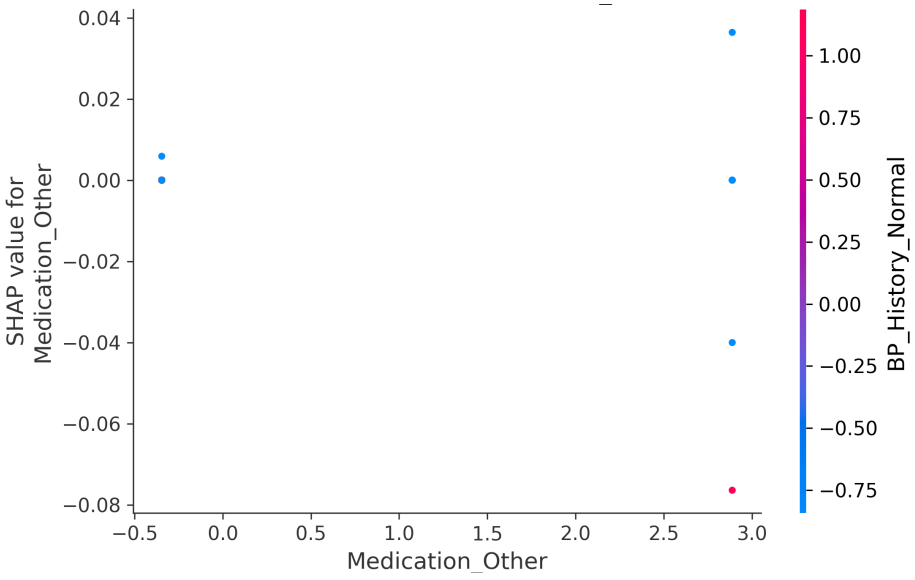
Na análise da Figura 29, revela a proeminência do histórico da pressão arterial como o fator mais decisivo para a predição do modelo. Com um SHAP médio de aproximadamente 0,49, seu impacto agregado é mais de três vezes superior ao da segunda *feature* mais influente, o *Family_History_Yes*. Fatores de risco secundários, como histórico familiar, tabagismo e idade, também demonstram alta relevância, seguidos por grupos relacionados ao estilo de vida, como o

Figura 27 – Dependência SHAP: *Medication Diuretic*.



Fonte: Autoria própria (2025).

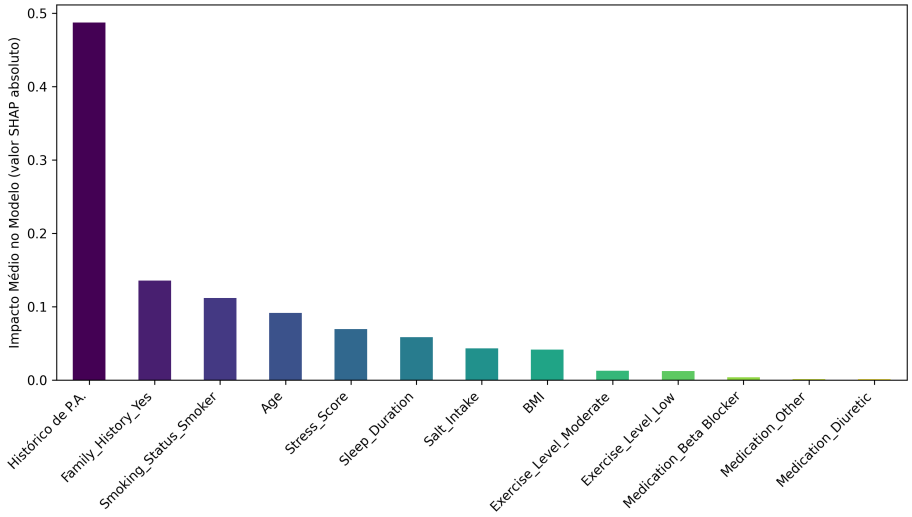
Figura 28 – Dependência SHAP: *Medication Other*.



Fonte: Autoria própria (2025).

nível de estresse e a duração do sono.

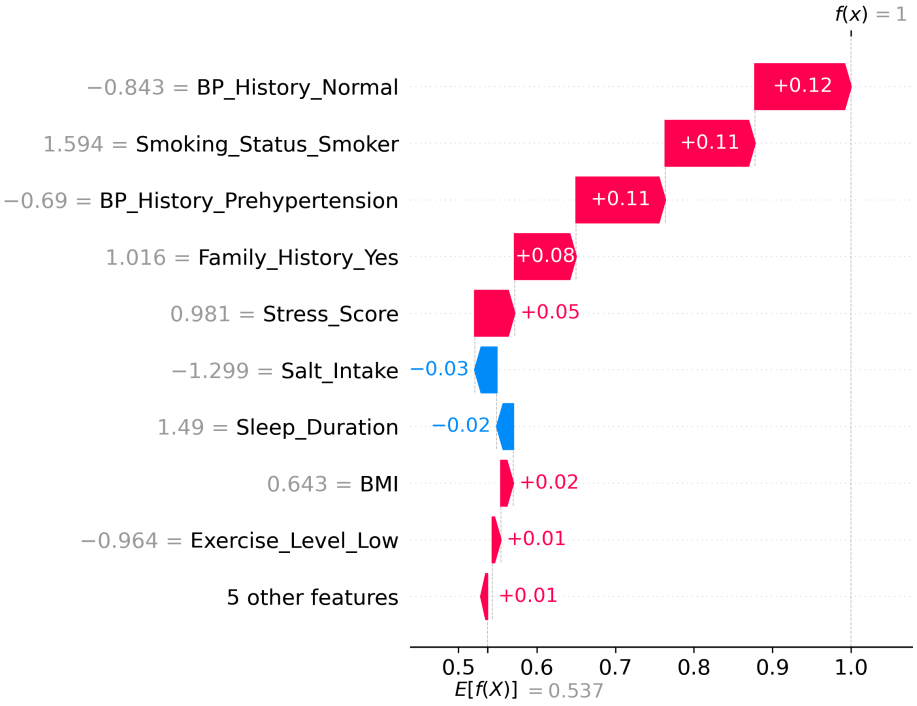
Figura 29 – Importância média das *features*.



Fonte: Autoria própria (2025).

Para a análise de como as *features* atuam em um paciente de forma exclusiva, os *Waterfall Plots* (Gráficos de cascata) dissecam predições individuais para entender o por quê o modelo chegou a uma conclusão específica. Na Figura 30, observamos o porquê o modelo previu um alto risco de hipertensão.

Figura 30 – Análise de predição individual - Alto Risco.



Fonte: Autoria própria (2025).

Na Figura 30, inferem-se os fatores que aumentam o risco de hipertensão, no qual os fatores que possuem a barra vermelha são os motivos pelos quais o modelo aumentou a probabilidade a partir da média. Já os fatores que diminuem o risco possuem barras azuis.

6 CONCLUSÕES

O presente trabalho teve como objetivo o desenvolvimento de um modelo de rede neural, *perceptron* multicamadas, e também de um modelo híbrido utilizando conceitos da mecânica quântica em uma das camadas ocultas. Além disso, realizamos uma análise comparativa entre os dois modelos utilizando de métricas conhecidas da literatura.

Inicialmente, para construirmos os algoritmos dos modelos, realizamos o tratamento dos dados e o pré-processamento antes de considerarmos construir as redes neurais. Considerando a utilização da arquitetura *perceptron multilayers*, possibilitando o uso de múltiplas camadas ocultas para treinamento e validação.

Mediante métricas, foi possível avaliar a eficiência dos modelos, avaliando o número de acertos e erros, precisão, *recall* e *F1-score*, além do comparativo entre a eficiência do tempo de execução entre o modelo clássico e híbrido. Também avaliamos, por meio da AI explicável, a influência de cada variável no risco de hipertensão.

Os resultados indicam que a integração de componentes quânticos melhorou o poder de generalização do modelo, ou seja, o modelo híbrido foi mais eficaz em aprender os padrões dos dados. Isso ocorre porque a utilização de conceitos da mecânica quântica, como a superposição e o entrelaçamento, processa informações em espaços de alta dimensionalidade, ou seja, o uso da computação quântica explora múltiplas soluções simultaneamente. Dessa forma, o modelo quântico híbrido tende a apresentar maior capacidade de generalização.

No entanto, em contrapartida, o modelo clássico obteve vantagem no seu tempo de execução, o que aponta uma limitação do modelo híbrido. Na análise das variáveis sobre hipertensão, observou-se que o maior impacto está associado à condição do histórico de pressão arterial. Isso ocorre porque o registro prévio de alterações na pressão, sejam valores elevados ou instáveis, reflete diretamente o comportamento fisiológico do sistema cardiovascular ao longo do tempo.

6.1 Trabalhos Futuros

Com base nos resultados obtidos, propõe-se, para trabalhos futuros, realizar testes em *hardwares* quânticos conhecidos, como o IBM Quantum da IBM, *Sycamore Processor*, também conhecido como Google Quantum AI da Google. Outro trabalho é realizar a aplicação de outras técnicas de regularização, como a L1 e L2, para melhorar os resultados do modelo. É analisado

também a possibilidade de aplicação do modelo em outra base de dados para fins de comparação dos modelos confeccionados neste trabalho.

O uso de otimização *bayesiana* para encontrar os melhores hiperparâmetros do modelo, como a taxa de aprendizado, número de neurônios, número de camadas, e etc., para obter resultados mais precisos, e a aplicação da técnica de validação cruzada variando o número de *K-Fold*. Esse método de validação permitiria uma avaliação mais robusta do desempenho do modelo, reduzindo a dependência de uma única divisão entre os conjuntos de treino e teste.

Outra contribuição seria a criação de um modelo com camadas ocultas inteiramente quânticas, para analisar e comparar com as camadas clássicas e híbridas. Para exploração da base de dados, a aplicação de modelos avançados de aprendizado de máquina frequentemente usados na literatura, como o *AdaBoost, Decision Tree, Random Forest, Gradient Boosting Machine* e entre outros. Por fim, aplicar o uso de computação quântica na construção de novos modelos de *deep learning*.

REFERÊNCIAS

ALI, S. *et al.* A comprehensive survey on brain tumor diagnosis using deep learning and emerging hybrid techniques with multi-modal mr image. **Archives of computational methods in engineering**, Springer, v. 29, n. 7, p. 4871–4896, 2022.

DIETTERICH, T. Overfitting and undercomputing in machine learning. **ACM computing surveys (CSUR)**, ACM New York, NY, USA, v. 27, n. 3, p. 326–327, 1995.

DIRAC, P. A. M. **The principles of quantum mechanics**. [S.l.]: Oxford university press, 1981.

DONATO, A. J.; MACHIN, D. R.; LESNIEWSKI, L. A. Mechanisms of dysfunction in the aging vasculature and role in age-related disease. **Circulation research**, Lippincott Williams & Wilkins Hagerstown, MD, v. 123, n. 7, p. 825–848, 2018.

DOND, V. T. Using predictive analytics to detect hypertension with machine learning. **INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT**, 2025.

FABIAN, P. Scikit-learn: Machine learning in python. **Journal of machine learning research 12**, p. 2825, 2011.

GABRICH, M. N.; FREITAS, H. C. de; SOUZA, M. A. Análise de redes neurais para crispr: Uma abordagem com computaçao quântica. In: SBC. **Simpósio em Sistemas Computacionais de Alto Desempenho (SSCAD)**. [S.l.], 2024. p. 13–24.

GERON, A. **Mãos à obra: aprendizado de maquina com Scikit-Learn, Keras & Tensorflow**. [S.l.]: Alta Books, 2021.

GOMIDE, F. A. **Redes neurais artificiais para engenharia e ciências aplicadas: curso prático**. [S.l.]: SciELO Brasil, 2012.

GONG, K.; CHEN, Y.; DING, X. Causal inference for hypertension prediction. In: IEEE. **2023 45th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)**. [S.l.], 2023. p. 1–4.

HARROW, A. W.; MONTANARO, A. Quantum computational supremacy. **Nature**, Nature Publishing Group, v. 549, n. 7671, p. 203–209, 2017.

KRZEMINSKA, J. *et al.* Arterial hypertension—oxidative stress and inflammation. **Antioxidants**, v. 11, 2022.

LUNDBERG, S. M.; LEE, S.-I. A unified approach to interpreting model predictions. **Advances in neural information processing systems**, v. 30, 2017.

MARIN, I.; GOGA, N. Hypertension detection based on machine learning. **Proceedings of the 6th Conference on the Engineering of Computer Based Systems**, 2019.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, Springer, v. 5, n. 4, p. 115–133, 1943.

MILLS, K. T.; STEFANESCU, A.; HE, J. The global epidemiology of hypertension. **Nature Reviews Nephrology**, Nature Publishing Group UK London, v. 16, n. 4, p. 223–237, 2020.

MÜLLER, A. C.; GUIDO, S. **Introduction to machine learning with Python: a guide for data scientists**. [S.l.]: "O'Reilly Media, Inc.", 2016.

NIELSEN, M. A.; CHUANG, I. L. **Quantum computation and quantum information**. [S.l.]: Cambridge university press, 2010.

PRESKILL, J. Quantum computing in the nisq era and beyond. **Quantum**, Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften, v. 2, p. 79, 2018.

SCHULD, M. *et al.* Circuit-centric quantum classifiers. **Physical Review A**, APS, v. 101, n. 3, p. 032308, 2020.

SHANKAR, R. **Principles of quantum mechanics**. [S.l.]: Springer Science & Business Media, 2012.

VANDERPLAS, J. **Python data science handbook: Essential tools for working with data**. [S.l.]: "O'Reilly Media, Inc.", 2016.

VAROQUAUX, G.; COLLIOT, O. Evaluating machine learning models and their diagnostic value. **Machine learning for brain disorders**, Springer, p. 601–630, 2023.

WES, M. **Python for data analysis**. [S.l.]: O'Reilly Publishing, 2013.

WILLIAMS, B. *et al.* 2018 esc/esh guidelines for the management of arterial hypertension: The task force for the management of arterial hypertension of the european society of cardiology (esc) and the european society of hypertension (esh). **European heart journal**, Oxford University Press, v. 39, n. 33, p. 3021–3104, 2018.

YING, X. An overview of overfitting and its solutions. In: IOP PUBLISHING. **Journal of physics: Conference series**. [S.l.], 2019. v. 1168, p. 022022.

ZHAO, H. *et al.* Predicting the risk of hypertension based on several easy-to-collect risk factors: A machine learning method. **Frontiers in Public Health**, v. 9, 2021.

APÊNDICE A – BIBLIOTECAS

Bibliotecas utilizadas para a construção dos modelos clássicos e híbridos e as versões utilizadas.

Biblioteca	**Versão**
numpy	1.26.4
pandas	2.2.2
python-dateutil	2.9.0.post0
pytz	2024.1
scikit-learn	1.4.2
scipy	1.13.0
six	1.16.0
threadpoolctl	3.5.0
tzdata	2024.1
imbalanced-learn	0.11.0
joblib	1.4.2
matplotlib	3.9.2
seaborn	0.12.2
tensorflow	2.15.0
pennylane	0.39.0
shap	0.46.0