



**Instituto Federal de Educação, Ciência e Tecnologia de São Paulo
Campus Sertãozinho**

Guilherme Santos da Silveira, Victoria de Oliveira Spagiari

**Aplicação de Sistemas Inteligentes para a
Detecção de Fraudes em Redes Elétricas**

Sertãozinho

2024

1 INTRODUÇÃO

1.1 Histórico da computação

2 REFERENCIAL TEÓRICO

2.1 Computação: fundamentos e evolução histórica

2.1.1 Definição geral de computação

De forma ampla, a computação pode ser definida como o conjunto de processos pelos quais um sistema recebe entradas, realiza algum tipo de processamento sobre essas informações e produz saídas ([KAISER; GRANADE, 2021](#)). Essa definição independe da natureza física do sistema computacional, sendo válida tanto para dispositivos mecânicos quanto eletrônicos ou digitais. O processamento pode envolver operações aritméticas, lógicas, simbólicas ou algorítmicas, desde que exista uma transformação bem definida entre os dados de entrada e os resultados obtidos na saída.

Sob essa perspectiva, a computação não se restringe aos computadores modernos, mas abrange qualquer mecanismo capaz de executar regras sistemáticas para manipulação de informação. Essa visão permite compreender a computação como um conceito anterior e mais amplo do que os próprios computadores digitais.

2.1.2 Computação antes dos computadores digitais

Muito antes do surgimento dos computadores eletrônicos, diferentes ferramentas e métodos foram desenvolvidos para auxiliar o ser humano em tarefas computacionais ([CAMPBELL-KELLY et al., 2023](#)). Um dos exemplos mais antigos é o ábaco, utilizado por diversas civilizações como instrumento para realizar operações aritméticas básicas. Embora simples, o ábaco já incorpora a ideia fundamental de representação e manipulação sistemática de informações.

Durante os séculos seguintes, surgiram dispositivos mecânicos mais sofisticados, como as máquinas de cálculo de Blaise Pascal e Gottfried Wilhelm Leibniz, capazes de executar automaticamente somas, subtrações e, em alguns casos, multiplicações e divisões. Paralelamente, grande parte dos cálculos científicos e de engenharia continuava sendo realizada manualmente, frequentemente com o auxílio de tabelas matemáticas e métodos numéricos.

Esses exemplos evidenciam que a computação, entendida como processamento de informação, antecede a computação digital moderna, sendo impulsionada por necessidades práticas nas áreas de comércio, ciência, engenharia e navegação.

2.1.3 Surgimento dos computadores digitais

O avanço tecnológico do século XX possibilitou a transição da computação mecânica para a computação eletromecânica e, posteriormente, eletrônica. Os primeiros computadores eletromecânicos utilizavam relés para implementar operações lógicas e aritméticas, enquanto os computadores eletrônicos passaram a empregar válvulas a vácuo, permitindo velocidades de processamento significativamente maiores.

Um exemplo histórico emblemático é o uso de máquinas computacionais no contexto da Segunda Guerra Mundial, especialmente nos esforços de criptoanálise. A máquina Enigma, utilizada para cifrar comunicações militares, representava um problema computacional de grande complexidade para a época ([HODGES, 2014](#)). O desenvolvimento de métodos sistemáticos e máquinas especializadas para quebrar suas cifras ilustra de forma clara como a computação passou a desempenhar um papel estratégico na resolução de problemas reais de alta relevância prática.

Esse período marca o início da computação digital como uma ferramenta central para lidar com problemas que extrapolavam a capacidade do cálculo manual ou puramente mecânico.

Após o surgimento dos primeiros computadores eletrônicos, a computação clássica passou por uma evolução acelerada, impulsionada principalmente por avanços no hardware. As válvulas a vácuo foram gradualmente substituídas por transistores, que ofereciam maior confiabilidade, menor consumo de energia e maior densidade de integração. Posteriormente, o desenvolvimento dos circuitos integrados permitiu a miniaturização e a integração de um número crescente de componentes em um único chip.

Esse avanço tecnológico é frequentemente associado à chamada Lei de Moore ([MOORE et al., 1965](#)), que observa o crescimento aproximadamente exponencial do número de transistores em circuitos integrados ao longo do tempo. Como consequência, houve um aumento contínuo no poder computacional disponível, tanto em termos de velocidade quanto de capacidade de armazenamento.

Além do aumento da densidade de componentes, a computação clássica também evoluiu por meio do paralelismo, explorando arquiteturas com múltiplos núcleos, processamento vetorial e sistemas distribuídos. Essas estratégias permitiram ganhos adicionais de desempenho, especialmente para aplicações que podem ser decompostas em tarefas paralelas.

2.1.4 Limitações da computação clássica

Apesar dos avanços expressivos, a computação clássica enfrenta limitações fundamentais quando aplicada a determinados tipos de problemas. Em muitos casos, o número de estados possíveis cresce exponencialmente com o tamanho do problema, tornando

inviável a exploração exaustiva de todas as soluções em tempo razoável.

A teoria da complexidade computacional formaliza essas limitações por meio de classes como P, NP, NP-difícil e NP-completo (SIPSER, 1996). Problemas pertencentes à classe P admitem soluções eficientes em tempo polinomial, enquanto problemas NP-completos, embora fáceis de verificar, não possuem algoritmos conhecidos que os resolvam eficientemente em computadores clássicos.

Exemplos práticos incluem problemas de otimização combinatória, como o problema do caixeiro viajante, a fatoração de grandes números inteiros e certos problemas de simulação de sistemas físicos complexos. Na prática, mesmo com hardware moderno e técnicas avançadas de paralelismo, esses problemas rapidamente se tornam intratáveis à medida que sua dimensão aumenta.

As limitações da computação clássica motivam a busca por novos paradigmas computacionais capazes de lidar de forma mais eficiente com problemas de alta complexidade. Em particular, há classes de problemas cuja estrutura sugere que abordagens baseadas exclusivamente em bits clássicos e operações determinísticas podem não ser suficientes.

Nesse contexto, surgem modelos alternativos de computação, como a computação quântica, que exploram princípios físicos distintos para representar e processar informações. Esses modelos não substituem a computação clássica, mas oferecem novas ferramentas conceituais e práticas para enfrentar problemas que “pedem” um novo tipo de tratamento computacional, abrindo caminho para avanços em áreas como criptografia, otimização e simulação de sistemas complexos.

2.2 Limites da computação clássica e motivação para novos paradigmas

2.2.1 Limitações tecnológicas versus limitações teóricas

Ao longo da história da computação, muitos obstáculos foram superados por meio de avanços tecnológicos, como o aumento da velocidade dos processadores, a miniaturização dos componentes eletrônicos e o aprimoramento das arquiteturas computacionais. Essas limitações, de natureza tecnológica, estão associadas a restrições práticas de engenharia e tendem a ser mitigadas com o progresso científico e industrial.

Porém, nem todas as limitações enfrentadas pela computação clássica podem ser superadas por avanços tecnológicos, existem barreiras de natureza teórica, intrínsecas ao próprio modelo computacional clássico, baseado na manipulação de bits e em operações determinísticas ou probabilísticas.

Essas limitações persistem mesmo quando se considera um computador idealizado

zado, capaz de executar operações em tempo arbitrariamente pequeno e com memória ilimitada. Nesses casos, o fator limitante não é a velocidade de cada operação individual, mas o crescimento do número total de operações necessárias para resolver determinados problemas.

Um exemplo paradigmático é o problema do logaritmo discreto ([SHOR, 1994](#)). Dado um grupo finito, um gerador g e um elemento h , o objetivo é encontrar o expoente x tal que $g^x = h$. Apesar de sua formulação simples, não existem algoritmos clássicos conhecidos que resolvam esse problema em tempo polinomial, mesmo considerando paralelismo clássico extremo. O melhor desempenho clássico conhecido ainda apresenta crescimento subexponencial, tornando o problema intratável para instâncias de grande porte. Assim, a dificuldade do logaritmo discreto ilustra um limite estrutural do modelo clássico de computação, e não apenas uma limitação tecnológica.

2.2.2 Crescimento exponencial do espaço de estados e exemplos computacionais

O crescimento exponencial do espaço de estados está no cerne de diversos problemas computacionais relevantes. Na teoria da complexidade, esse fenômeno aparece de forma clara em problemas pertencentes às classes NP-difícil e NP-completo ([SIPSER, 1996](#)), nos quais o número de soluções candidatas cresce combinatorialmente com o tamanho do problema.

O problema do caixeiro viajante (*Traveling Salesman Problem – TSP*) é um exemplo clássico. Dado um conjunto de cidades e as distâncias entre elas, o objetivo é encontrar o caminho de menor custo que visite cada cidade exatamente uma vez. A versão de decisão do problema é NP-completa, e a versão de otimização é NP-difícil. Na prática, instâncias de grande porte são tratadas por meio de heurísticas e algoritmos aproximados, como algoritmos genéticos ou *simulated annealing* ([KIRKPATRICK; JR; VECCHI, 1983](#)), que produzem boas soluções, mas sem garantia de optimidade global.

Outro exemplo de grande relevância é o problema da fatoração de números inteiros grandes, base de sistemas criptográficos como o RSA ([RIVEST; SHAMIR; ADLEMAN, 1978](#)). Embora não se saiba se esse problema é NP-completo, ele não admite algoritmos clássicos eficientes conhecidos. Nesse caso, a computação quântica oferece uma vantagem estrutural: o algoritmo de Shor ([SHOR, 1994](#)) demonstra que a fatoração pode ser realizada em tempo polinomial em um computador quântico, evidenciando que o obstáculo não está apenas no hardware, mas no modelo computacional utilizado.

Essas dificuldades tornam-se ainda mais pronunciadas na simulação de sistemas físicos quânticos. Um sistema composto por n partículas quânticas requer a descrição de um estado em um espaço de dimensão proporcional a 2^n . Como consequência, a simulação

exata desses sistemas em computadores clássicos torna-se rapidamente inviável, mesmo para valores modestos de n , reforçando a ideia de que certos problemas apresentam uma incompatibilidade fundamental com o paradigma clássico de computação.

2.2.3 Computação não-clássica e a motivação física de Feynman

Diante dessas limitações estruturais, surgem propostas de paradigmas de computação não-clássica, que buscam explorar modelos físicos alternativos para representar e processar informação. A computação analógica utiliza grandezas contínuas e a dinâmica natural de sistemas físicos para realizar cálculos, enquanto a computação neuromórfica inspira-se na organização do cérebro, explorando paralelismo massivo e adaptação.

Entre esses paradigmas, a computação quântica se destaca por sua fundamentação direta nas leis da mecânica quântica. Em vez de tentar simular sistemas quânticos por meio de bits clássicos, a computação quântica propõe utilizar sistemas quânticos controlados como elementos computacionais, explorando propriedades como superposição, interferência e emaranhamento.

Essa ideia foi articulada de forma pioneira por Richard Feynman ([FEYNMAN, 2018](#)), ao observar que a dificuldade de simular sistemas quânticos em computadores clássicos não é acidental, mas resulta do fato de que a natureza não opera segundo as regras da computação clássica. Assim, a computação quântica surge não apenas como uma alternativa tecnológica, mas como um novo paradigma conceitual, alinhado à própria estrutura física dos sistemas que se deseja estudar.

2.3 Fundamentos da computação quântica

2.3.1 Qubits

O bit quântico (*quantum bit*), ou simplesmente qubit, é a unidade fundamental da computação quântica, assim como o bit é para a computação clássica. Comparar um bit clássico com um qubit ajuda a compreender sua natureza. Um bit clássico pode assumir apenas um de dois valores possíveis: 0 ou 1. Enquanto nada o modifique, ele permanecerá em um desses dois estados fixos ([SUTOR, 2019](#)).

De maneira análoga, quando medimos um qubit, obtemos um de dois resultados possíveis: $|0\rangle$ ou $|1\rangle$ (a notação de *bras* e *kets* será abordada em capítulo posterior). A principal diferença é que, antes da medição, o qubit pode existir em um estado de superposição, descrito pela combinação linear:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad \text{com } |\alpha|^2 + |\beta|^2 = 1,$$

onde α e β são amplitudes complexas associadas às probabilidades de o qubit ser encontrado em $|0\rangle$ ou $|1\rangle$, respectivamente (SUTOR, 2019).

A superposição é um princípio fundamental da mecânica quântica que estabelece que, antes da medição, um sistema pode existir simultaneamente em múltiplos estados possíveis. No ato da medição, ocorre o chamado *colapso da função de onda*, fazendo o sistema assumir um dos estados definidos. No caso de um qubit, os dois estados possíveis formam uma base bidimensional, e a superposição pode ser visualizada geometricamente na Esfera de Bloch (Fig. 1), que representa todos os estados quânticos possíveis de um qubit.

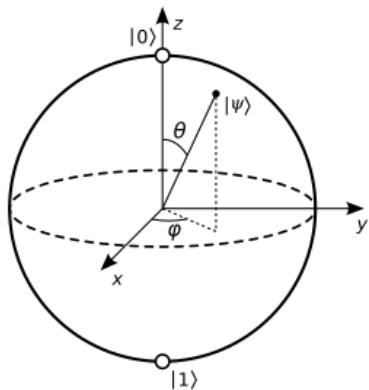


Figura 1 – Representação de um qubit na Esfera de Bloch.

Um equívoco comum é atribuir à superposição, isoladamente, a origem do alto poder computacional dos computadores quânticos. Embora ela seja condição necessária, não é suficiente. A superposição permite o *paralelismo quântico*, em que múltiplas combinações de estados podem ser processadas simultaneamente. No entanto, sem mecanismos de interferência controlada, as medições resultariam apenas em resultados aleatórios, não necessariamente úteis para resolver um problema.

É por meio da interferência entre amplitudes quânticas que os algoritmos quânticos exploram as propriedades da superposição para eliminar resultados incorretos e reforçar os corretos. Os algoritmos de Shor e de Grover exemplificam esse princípio: ambos utilizam a superposição e a interferência de forma controlada para atingir ganhos de desempenho significativos em relação aos métodos clássicos.

2.4 Conceitos complementares

Essa seção se dedica a explicar conceitos que formam a base da computação quântica

2.5 Brakets

Na computação quântica é extremamente comum o uso de vetores e matriz para a representação dos estados quânticos. De forma geral podemos ter vetores em forma de linha e em forma de coluna, vamos considerar o vetor $\mathbf{v} = v_1, v_2, \dots, v_n$

3 METODOLOGIA

O objetivo deste trabalho é treinar e aplicar o método inteligente *Support Vector Machine* (SVM) (??) para identificar clientes fraudulentos em um banco de dados de uma companhia distribuidora de energia elétrica. O SVM será treinado para diferenciar entre padrões de consumo legítimos e fraudulentos, auxiliando na detecção de fraudes com maior precisão.

A pesquisa adota uma abordagem quantitativa, focada na análise de dados históricos fornecidos por uma companhia distribuidora de energia elétrica e disponibilizados gratuitamente na plataforma *Kaggle* (??). Essa abordagem permite a aplicação de técnicas de aprendizado de máquina para modelagem preditiva e identificação de padrões associados ao furto de energia elétrica.

O desenvolvimento do projeto será conduzido utilizando a linguagem de programação *Python*, escolhida por sua robustez e extensa biblioteca de ferramentas para análise de dados e aprendizado de máquina. O método *Support Vector Machine* será implementado com a biblioteca *Scikit-Learn* (??). A metodologia possui as seguintes etapas: tratamento de dados, treinamento do modelo, avaliação da assertividade e escrita do trabalho.

Inicialmente, será realizado o tratamento do banco de dados com informações do consumo de energia, uma vez que é comum que esses dados apresentem problemas como informações faltantes, valores discrepantes e outras inconsistências. Essa etapa é essencial para garantir a qualidade do treinamento do modelo.

Após o tratamento, o banco de dados será dividido em duas partes: 80% dos dados serão utilizados para o treinamento do SVM, enquanto os 20% restantes serão reservados para a validação do modelo, cuja eficácia será avaliada utilizando o coeficiente de determinação R^2 (????).

Além disso, será realizada uma Análise Multivariada de Variância (MANOVA) (??????) para entender melhor os principais fatores que influenciam o furto de energia elétrica, permitindo uma análise mais detalhada dos dados e dos padrões detectados.

Por fim, a escrita da monografia será realizada com base nos dados e resultados obtidos ao longo do projeto. Para garantir a qualidade do texto, será utilizada uma ferramenta de Inteligência Artificial para correções ortográficas e ajustes gramaticais, proporcionando maior precisão e clareza na apresentação dos resultados.

REFERÊNCIAS

- CAMPBELL-KELLY, M. et al. **Computer: A history of the information machine.** [S.l.]: Routledge, 2023. Citado na página [2](#).
- FEYNMAN, R. P. Simulating physics with computers. In: **Feynman and computation.** [S.l.]: cRc Press, 2018. p. 133–153. Citado na página [6](#).
- HODGES, A. **Alan Turing: The Enigma: The Book That Inspired the Film "The Imitation Game".** [S.l.]: Princeton University Press, 2014. Citado na página [3](#).
- KAISER, S. C.; GRANADE, C. **Learn quantum computing with python and Q#: A hands-on approach.** [S.l.]: Simon and Schuster, 2021. Citado na página [2](#).
- KIRKPATRICK, S.; JR, C. D. G.; VECCHI, M. P. Optimization by simulated annealing. **science**, American association for the advancement of science, v. 220, n. 4598, p. 671–680, 1983. Citado na página [5](#).
- MOORE, G. E. et al. **Cramming more components onto integrated circuits.** 1965. Citado na página [3](#).
- RIVEST, R. L.; SHAMIR, A.; ADLEMAN, L. A method for obtaining digital signatures and public-key cryptosystems. **Communications of the ACM**, ACM New York, NY, USA, v. 21, n. 2, p. 120–126, 1978. Citado na página [5](#).
- SHOR, P. W. Algorithms for quantum computation: discrete logarithms and factoring. In: **IEEE. Proceedings 35th annual symposium on foundations of computer science.** [S.l.], 1994. p. 124–134. Citado na página [5](#).
- SIPSER, M. Introduction to the theory of computation. **ACM Sigact News**, ACM New York, NY, USA, v. 27, n. 1, p. 27–29, 1996. Citado 2 vezes nas páginas [4](#) e [5](#).
- SUTOR, R. **Dancing with qubits.** [S.l.]: Packt Publishing Birmingham, UK, 2019. v. 1. Citado 2 vezes nas páginas [6](#) e [7](#).