



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Murilo Araujo Souza

**Detecção de Perdas Não Técnicas em Infraestruturas Avançadas de Medição: Uma
Abordagem Através de Aprendizagem de Máquina**

Recife

2025

Murilo Araujo Souza

**Detecção de Perdas Não Técnicas em Infraestruturas Avançadas de Medição: Uma
Abordagem Através de Aprendizagem de Máquina**

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

Área de Concentração: Processamento de Energia.

Orientador: Prof. Dr. Ronaldo Ribeiro Barbosa de Aquino

Coorientadora: Prof.^a Dra. Aida Araújo Ferreira

Recife

2025

.Catalogação de Publicação na Fonte. UFPE - Biblioteca Central

Souza, Murilo Araújo.

Detecção de perdas não técnicas em infraestruturas avançadas de medição: uma abordagem através de aprendizagem de máquina / Murilo Araújo Souza. - Recife, 2025.

84f.: il.

Tese (Doutorado) - Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, Programa de Pós-Graduação em Engenharia Elétrica, 2025.

Orientação: Ronaldo Ribeiro Barbosa de Aquino.

Coorientação: Aida Araújo Ferreira.

Inclui referências.

1. Perdas não técnicas; 2. Aprendizagem de máquina; 3. Extração de atributos; 4. Multilayer perceptron; 5. Long short-term memory. I. Aquino, Ronaldo Ribeiro Barbosa de. II. Ferreira, Aida Araújo. III. Título.

UFPE-Biblioteca Central

AGRADECIMENTOS

Gostaria de agradecer imensamente ao professor Ronaldo Aquino por todos os momentos de ensino e conhecimento compartilhados durante todo o doutorado. Desde o início, ele se mostrou extremamente paciente ao ouvir todas as ideias que tive e orientar da melhor maneira possível os rumos a serem tomados. Também devo agradecer por todo o apoio e confiança que me foram dados para realizar outras atividades que auxiliaram na minha formação.

À professora Aida Ferreira, por todas as sugestões que definiram o rumo final que este trabalho tomou.

A todos os amigos do LDSP, pela amizade, companhia, momentos de descontração e, inclusive, pelas trocas de informações sobre os mais diversos assuntos. A presença de todos vocês sempre tornou o ambiente no laboratório muito mais acolhedor.

A meus pais, Antônio e Nelma, e aos meus irmãos, Marcelo e Milena, que, pela terceira vez, me deram todo o suporte e incentivo necessários para que eu continuasse os meus estudos e alcançasse mais esse objetivo. Novamente, sou grato por compreenderem minha ausência durante longos períodos. Muito obrigado pelo apoio incondicional. Com vocês, eu cheguei até aqui. Um agradecimento mais que especial à minha sobrinha Alice, que, apesar de eu estar ausente durante tanto tempo, sempre lembra de mim e, assim que retorno, está sempre feliz por me ver e começar suas brincadeiras.

À família Brasil, principalmente à Mayara, por todo o carinho dado durante todos esses anos, me fazendo sentir acolhido, em casa. Todas as festas, passeios e reuniões familiares foram extremamente agradáveis.

À CAPES, pela concessão da bolsa de estudos para que este trabalho pudesse ser concluído de maneira satisfatória.

Muito obrigado!

RESUMO

Perdas de energia ocorrem em qualquer sistema elétrico, sejam elas devido à fenômenos físicos ou à ação humana. Essas perdas podem ser técnicas, aquelas inerentes ao sistema, e não técnicas, também conhecidas como perdas comerciais, geralmente associadas a algum tipo de fraude cometido pelo consumidor. Algoritmos de Aprendizagem de Máquina podem ser utilizados para detectar padrões de consumo de energia elétrica, afim de identificar consumidor que eventualmente esteja cometendo fraude. Através de dados reais de consumo de energia elétrica de consumidores honestos, foram empregadas equações matemáticas para modelar onze tipos de perdas não técnicas que podem ocorrer em um medidor inteligente de energia elétrica contido em uma infraestrutura avançada de medição. Além disso, baseado nas séries temporais de consumo de energia, foram extraídos atributos estatísticos (média e variância), temporais (coordenadas do centroide) e de redução de dimensionalidade (*Uniform Manifold Approximation and Projection*) para aumentar significativamente o desempenho de classificação do modelo. O modelo proposto é do tipo multientradas, com dois ramos em paralelo: um contendo uma rede *Multilayer Perceptron* e o outro uma rede *Long Short-Term Memory* que processam diferentes informações. As saídas dessas redes são combinadas e fornecidas como entrada para outra *Multilayer Perceptron* que realiza a classificação final. As simulações foram realizadas no IDE *Spyder* através da linguagem de programação *Python*. Os resultados demonstram o impacto positivo de cada atributo no aprimoramento da classificação. Em particular, os atributos derivados da técnica *Uniform Manifold Approximation and Projection* comprovam a eficácia desse método quando integrado ao modelo proposto. Este modelo, por sua vez, superou abordagens anteriores para a mesma finalidade, atingindo maior eficácia na identificação de perdas não técnicas.

Palavras-chave: perdas não técnicas; aprendizagem de máquina; extração de atributos; *multilayer perceptron*; *long short-term memory*.

ABSTRACT

Energy losses occur in any electrical system, whether due to physical phenomena or human actions. These losses can be technical, inherent to the system, and non-technical, also known as commercial losses, usually associated with some type of fraud committed by the consumer. Machine Learning algorithms can be used to detect electricity consumption patterns in order to identify consumers who may be committing fraud. Using real electricity consumption data from honest consumers, mathematical equations were employed to model eleven types of non-technical losses that can occur in a smart electricity meter contained in an advanced metering infrastructure. Furthermore, based on electricity consumption time series, statistical (mean and variance), temporal (centroid coordinates), and dimensionality reduction (*Uniform Manifold Approximation and Projection*) features were extracted to significantly enhance the classification performance of the model. The proposed model is a multi-input type with two parallel branches: one containing a *Multilayer Perceptron* network and the other a *Long Short-Term Memory* network that process different information. The outputs of these networks are combined and provided as input to another *Multilayer Perceptron* that performs the final classification. The simulations were carried out in the *Spyder* IDE using the *Python* programming language. The results demonstrate the positive impact of each feature on classification performance. In particular, the features derived from the *Uniform Manifold Approximation and Projection* technique confirm the effectiveness of this method when integrated with the proposed model. This model, in turn, outperformed previously developed models for the same purpose, achieving greater effectiveness in identifying non-technical losses.

Keywords: non-technical losses; machine learning; feature extraction; multilayer perceptron; long short-term memory.

LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplos de perdas técnicas e não técnicas.	17
Figura 2 – Fluxo de potência no SEP.	18
Figura 3 – Percentuais de perdas em relação a energia injetada na rede de distribuição no Brasil em 2023.	23
Figura 4 – PNTR e PNTReg nos últimos anos no Brasil sobre o mercado de baixa tensão.	25
Figura 5 – Custo da PNTR no Estado de Pernambuco.	26
Figura 6 – Custo da PNTR no Brasil.	27
Figura 7 – Aprendizagem por Reforço.	34
Figura 8 – Tipos de Aprendizagem de Máquina.	34
Figura 9 – Problema de classificação binário.	35
Figura 10 – Problema de classificação multiclases.	35
Figura 11 – Problema de regressão.	36
Figura 12 – Neurônio biológico.	37
Figura 13 – Neurônio MCP.	37
Figura 14 – Representação do MLP.	40
Figura 15 – Estrutura básica de uma RNN.	43
Figura 16 – RNN básica desdobrada ao longo do tempo.	44
Figura 17 – Célula da LSTM.	46
Figura 18 – LSTM desdobrada ao longo do tempo.	46
Figura 19 – Dados em dois agrupamentos distintos.	53
Figura 20 – Redução de dimensionalidade.	54
Figura 21 – Inicialização dos pontos na dimensão inferior.	56
Figura 22 – Similaridade na dimensão inferior.	56
Figura 23 – Consumo real e fraude do tipo $f = 1$	58
Figura 24 – Consumo real e fraude do tipo $f = 10$	58
Figura 25 – Consumo real e fraude do tipo $f = 10$ com suas coordenadas dos centroides.	59
Figura 26 – Redução de dimensionalidade de um consumidor através do UMAP.	60
Figura 27 – Redução de dimensionalidade de dez consumidores através do UMAP.	61
Figura 28 – Rede modular.	64
Figura 29 – Matriz de confusão binária.	65
Figura 30 – Matriz de confusão para 3 classes.	65

Figura 31 – Validação cruzada <i>k-fold</i>	67
Figura 32 – Métricas por classe para o modelo M16.	72
Figura 33 – Matriz de confusão para o modelo M16.	73
Figura 34 – Consumo total de 100 consumidores sem fraudes ao longo de 30 dias. . . .	74
Figura 35 – Consumo total de 100 consumidores que cometeram algum tipo de fraude ao longo de 30 dias.	74
Figura 36 – Matriz de confusão para modelo M16 referente à base de 100 consumidores.	75

LISTA DE TABELAS

Tabela 1 – Percentuais de PNTs sobre a energia injetada por região em 2023.	24
Tabela 2 – Percentuais de PNTR por região sobre o mercado de baixa tensão.	25
Tabela 3 – Valores da TME e de PNTR no Estado de Pernambuco.	26
Tabela 4 – Funções de ativação comumente utilizadas na literatura.	38
Tabela 5 – Modelos desenvolvidos por Souza <i>et al.</i> (2024).	63
Tabela 6 – Novos modelos propostos.	63
Tabela 7 – Parâmetros de treinamento.	66
Tabela 8 – Resultados dos modelos M1 e M2.	68
Tabela 9 – Resultados dos modelos M3, M4, M5 e M6.	68
Tabela 10 – Tempo de treinamento dos modelos M3, M4, M5 e M6.	69
Tabela 11 – Resultados dos modelos M7 e M8.	69
Tabela 12 – Resultados dos modelos M9 até M17.	70
Tabela 13 – Resultados da validação cruzada para o modelo M16.	71
Tabela 14 – Receitas e prejuízo financeiro causado sem aplicação do modelo M16.	75

LISTA DE ABREVIATURAS E SIGLAS

ADASYN	<i>Adaptive Synthetic Sampling</i> (Amostragem Sintética Adaptativa)
AM	Aprendizagem de Máquina
AMI	<i>Advanced Metering Infrastructure</i> (Infraestrutura Avançada de Medição)
ANEEL	Agência Nacional de Energia Elétrica
BGD	<i>Batch Gradient Descent</i> (Gradiente Descendente por Lotes)
CER	<i>Commission for Energy Regulation</i> (Comissão de Regulamentação Energética)
CI	Circuito Integrado
CNN	<i>Convolutional Neural Network</i> (Rede Neural Convolucional)
DL	<i>Deep Learning</i> (Aprendizagem Profunda)
CA\$	Dólar Canadense
US\$	Dólar Estadunidense
FA	<i>Firefly Algorithm</i> (Algoritmo do vaga-lume)
FN	Falso Negativo
FP	Falso Positivo
FS	<i>Feature Selection</i> (Seleção de Atributos)
GD	Gradiente Descendente
HS	<i>Harmony Search</i> (Busca Harmônica)
IDE	<i>Integrated Development Environment</i> (Ambiente de Desenvolvimento Integrado)
IA	Inteligência Artificial
ISSDA	<i>Irish Social Science Data Archive</i> (Arquivo de Dados de Ciências Sociais Irlandês)

kNN	<i>k-Nearest Neighbors</i> (k-Vizinhos mais Próximos)
kVA	Kilovolt-Ampère
kW	Kilowatt
LSTM	<i>Long Short-Term Memory</i> (Memória de Curto e Longo Prazo)
MBGD	<i>Mini-Batch Gradient Descent</i> (Gradiente Descendente por Mini-lotes)
MCP	Modelo Mcculloch-Pitts
MLP	<i>Multilayer Perceptron</i> (Perceptron Multicamadas)
OPF	<i>Optimum-Path Forest</i> (Floresta de Caminhos Ótimos)
PCA	<i>Principal Component Analysis</i> (Análise de Componentes Principais)
PNT	Perda Não Técnica
PNTR	Perda Não Técnica Real
PNTReg	Perda Não Técnica Regulatória
PS	Pontuação de similaridade
PSO	<i>Particle Swarm Optimization</i> (Enxame de Partículas)
PT	Perda Técnica
QEE	Qualidade de Energia Elétrica
R\$	Real
RNA	Redes Neurais Artificiais
RNN	Rede Neural Recorrente
SEP	Sistema Elétrico de Potência
SG	<i>Smart Grids</i> (Redes Elétricas Inteligentes)
SGD	<i>Stochastic Gradient Descent</i> (Gradiente Descendente Estocástico)
SMOTE	<i>Synthetic Minority Over-sampling Technique</i> (Técnica de Sobreamostragem Minoritária Sintética)

TSF	<i>Time Series Forest</i>
MCDCNN	<i>Multi Channel Deep Convolutional Neural Network</i>
SOM	<i>Self-Organizing Map</i> (Mapa Auto-Organizável de Kohonen)
SVM	<i>Support Vector Machine</i> (Máquina de Vetores de Suporte)
TC	Transformador de Corrente
TME	Tarifa Média de Energia
TN	<i>True Negative</i> (Verdadeiro Negativo)
TP	<i>True Positive</i> (Verdadeiro Positivo)
t-SNE	<i>t-Distributed Stochastic Neighbor Embedding</i>
UMAP	<i>Uniform Manifold Approximation and Projection</i>

LISTA DE SÍMBOLOS

E_G	Energia total gerada
E_{ST}	Energia que flui pelo sistema de transmissão
P_{ST}	Perdas técnicas no sistema de transmissão
E_{SD}	Energia que flui pelo sistema de distribuição
P_{SD}	Perdas técnicas no sistema de distribuição
E_{TC}	Energia total entregue ao consumidor
E_{NF}	Energia não faturada pela concessionária
E_F	Energia faturada pela concessionária
θ	Limiar do neurônio MCP
\mathbf{x}	Vetor de variáveis de entrada (atributos)
\mathbf{w}	Vetor de pesos sinápticos
μ	Função de ativação
z	Produto interno entre \mathbf{x} e \mathbf{w}
T	Número de atributos/instantes de tempo na base de dados
t	t -ésimo atributo em uma base de dados
m	Número de nós na camada de saída de uma rede neural
w_0	<i>Bias</i>
e	Erro entre a saída desejada e a saída real de um neurônio
\mathbf{d}	Vetor de saídas desejadas (rótuladas)
η	Taxa de aprendizagem
ψ	Termo de momento
ψ_α	Constante do termo de momento ψ

ω_x	Peso sináptico associados à entrada de uma RNN
ω_y	Peso sináptico associados à saída de uma RNN
ω_s	Peso sináptico associados ao estado oculto de uma RNN
S_t	Estado de uma rede recorrente no instante t (referente ao t -ésimo atributo)
\mathbf{y}	Vetor de variáveis de saída (previsões)
b_s	<i>Bias</i> do estado oculto de uma RNN
b_y	<i>Bias</i> do neurônio de saída de uma RNN
f_t	Valor de entrada da porta de esquecimento de uma LSTM no instante t
i_t	Valor de entrada do primeiro termo da porta de entrada de uma LSTM no instante t
g_t	Valor de entrada do segundo termo da porta de entrada de uma LSTM no instante t
o_t	Valor de entrada porta de saída de uma LSTM no instante t
y_t	Valor de saída de uma LSTM no instante t
C	Número de consumidores na base de dados
D	Número de dias na base de dados
N	Número de amostras total da base de dados
\mathbf{X}	Matriz contendo as séries temporais de consumo de energia de todos os consumidores
\mathbf{m}_f	Vetor com tipo f de amostra maliciosa
α	Valor aleatório entre 0,1 e 0,9
t_s	Horário inicial de um ataque ao medidor inteligente de energia
t_e	Horário final de um ataque ao medidor inteligente de energia
β	Variável que zera o consumo de energia entre os instantes t_s e t_e

γ_t	Vetor de valores aleatórios entre 0,1 e 0,9 em cada instante de tempo t distinto
κ	Variável que indica o contorno parcial do consumo de energia entre os instantes t_s e t_e
co	Ponto de corte
s	Taxa de variação do ataque tipo 8
i_{max}	Intensidade máxima do ataque tipo 8
i_{min}	Intensidade mínima do ataque tipo 8
f_r	Fator de redução referente ao ataque do tipo 11
N_a	Número de instantes de tempo em que o ataque do tipo 11 ocorre
δ_E	Quantidade total de energia que foi reduzida durante um ataque do tipo 11
τ	Dimensão dos dados no espaço reduzido através do algoritmo UMAP
ρ_i	Conectividade local do ponto i do algoritmo UMAP
σ_i	Fator de suavização do ponto i da curva exponencial do algoritmo UMAP
p	Pontuação de similaridade do algoritmo UMAP na dimensão original
q	Pontuação de similaridade do algoritmo UMAP na dimensão inferior (<i>embedding</i>)
δ	Distância desejada entre um par de pontos na dimensão inferior
C	Função de custo do algoritmo UMAP
P	Precisão
R	<i>Recall</i>

SUMÁRIO

1	INTRODUÇÃO	16
1.1	ABORDAGEM DO PROBLEMA E JUSTIFICATIVA	19
1.2	OBJETIVOS DA PESQUISA	20
1.2.1	Objetivos Específicos	20
1.3	ORGANIZAÇÃO DO TEXTO	20
2	REVISÃO DA LITERATURA	22
2.1	PNT NO BRASIL	23
2.1.1	PNT Real e Regulatória	24
2.2	TRABALHOS RELEVANTES	27
3	APRENDIZAGEM DE MÁQUINA	33
3.1	PERCEPTRON MULTICAMADAS - MLP	36
3.1.1	Treinamento do MLP	40
3.2	<i>LONG SHORT-TERM MEMORY</i> - LSTM	43
4	METODOLOGIA	48
4.1	BASE DE DADOS	48
4.2	MODELAGEM DAS PNT	49
4.3	REDUÇÃO DE DIMENSIONALIDADE	52
4.3.1	<i>Uniform Manifold Approximation and Projection</i> - UMAP	53
4.3.2	Extração de Atributos	57
4.4	MODELOS DESENVOLVIDOS	61
4.5	MÉTRICAS DE AVALIAÇÃO	64
4.6	METODOLOGIA DE AVALIAÇÃO	66
5	DISCUSSÃO DOS RESULTADOS	68
6	CONCLUSÕES	77
6.1	SUGESTÕES DE TRABALHOS FUTUROS	78
	REFERÊNCIAS	80

1 INTRODUÇÃO

O Sistema Elétrico de Potência (SEP) é composto por usinas de geração, linhas de transmissão em alta tensão e sistemas de distribuição em média e baixa tensão. As usinas de geração são instaladas próximas aos locais onde se possa aproveitar os recursos energéticos, como, por exemplo, hidroelétricas instaladas em quedas d'água de rios ou locais que tendem a formar lagos (Zanetta Junior, 2006) e painéis solares em locais que possuam elevada incidência de radiação solar. Logo após a geração, ocorre a etapa de transmissão, que consiste no processo de elevar o nível de tensão gerado para que seja transmitida à longas distâncias através das linhas de transmissão. Por fim, as linhas de transmissão chegam próximas aos grandes centros de consumo, onde começa a etapa da distribuição de energia, em que o nível de tensão deve ser novamente reduzido a valores mais baixos para que possa ser utilizada pelo consumidor final.

Nem toda energia gerada é entregue totalmente ao consumidor final, ou seja, existem perdas de energia elétrica ocorrendo durante todas as etapas do SEP. Essas perdas são divididas em perdas técnicas (PT) e não técnicas (PNT), e é no setor de distribuição que elas possuem o valor mais elevado em todo o SEP (Navani; Sharma; Sapra, 2012).

As PT são inerentes a um sistema, pois elas são resultantes de fenômenos físicos (Rodrigues *et al.*, 2021). Segundo Sallam e Malik (2011), elas ainda podem ser subdivididas em perdas variáveis (comumente chamadas de perdas no cobre) e perdas fixas (comumente chamadas de perdas no núcleo do transformador). Estas são as perdas por histerese e por correntes de Foucault, enquanto que aquelas estão relacionadas ao aquecimento dos condutores devido à passagem de corrente elétrica. Como são provenientes de fenômenos físicos, as perdas técnicas não podem ser totalmente eliminadas, porém, ao longo dos anos, diversos estudos vêm sendo realizados com o intuito de minimizar esse tipo de perda (Gauche, 1994; Christiano; Vargas; Zuben, 2003; Barbosa *et al.*, 2021).

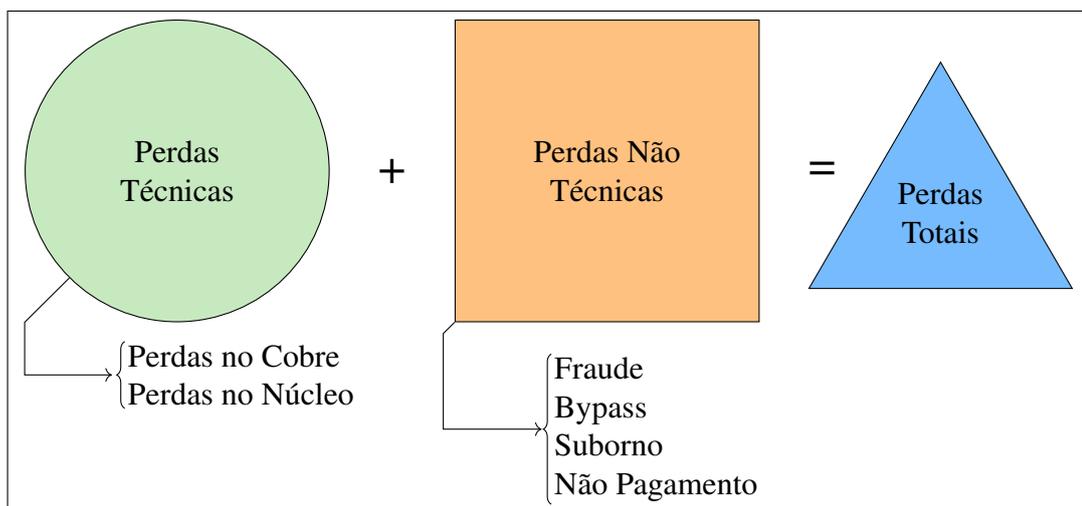
As PNT, também conhecidas na literatura como perdas comerciais (Ferreira, 2008; Ramos, 2014; Arango, 2019), são aquelas unidades de energia entregues e consumidas, mas que não são faturadas pela concessionária de energia elétrica. De acordo com Smith (2004), existem quatro formas para contabilizar uma PNT, a saber:

- a) **Fraude:** Ocorre quando o consumidor deliberadamente tenta enganar a concessionária. A fraude geralmente é cometida através de adulteração dos medidores de energia, fazendo com que esses registrem valores inferiores de consumo de energia elétrica ao que realmente ocorreu;

- b) **Bypass:** O *bypass* é uma ligação clandestina realizada direto da fonte (rede de baixa tensão, por exemplo) para a carga, sem passar pelo medidor de energia;
- c) **Suborno:** O suborno dos leituristas da concessionária é outra prática comum. A corrupção pode partir tanto do consumidor quanto do leiturista;
- d) **Não Pagamento:** É o simples fato do consumidor ignorar a conta de energia, não efetuando o seu pagamento. Em decorrência do consumidor não residir mais no local ou porque uma empresa foi a falência.

Na Figura 1 são retratados os exemplos de perdas de energia elétrica. Assim, o somatório das PT e das PNT totalizam as perdas totais em um sistema.

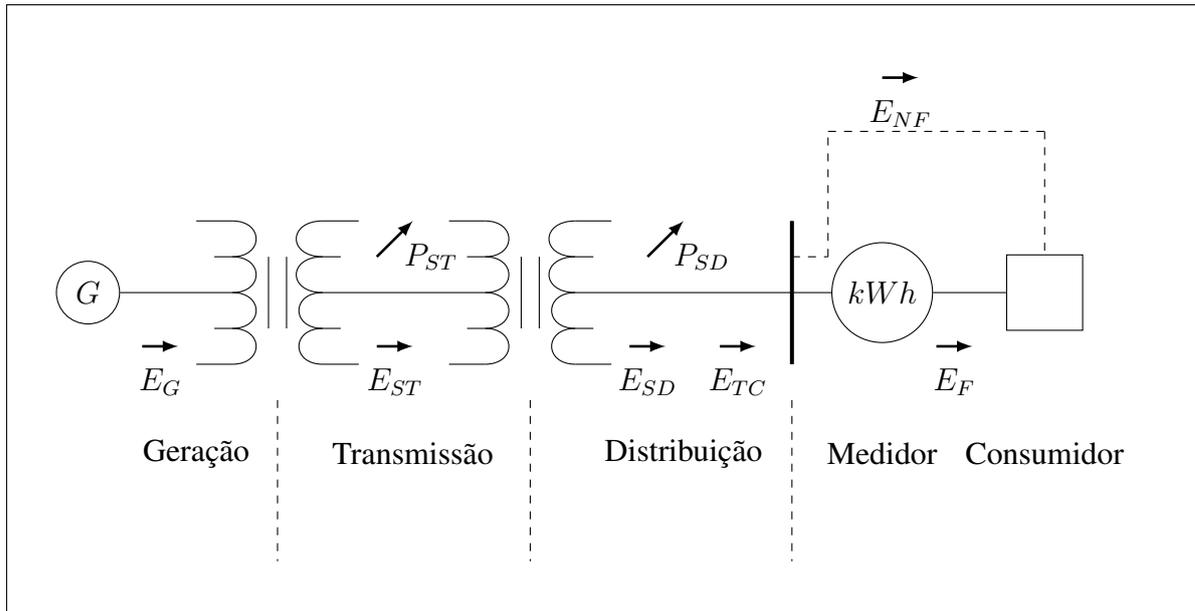
Figura 1 – Exemplos de perdas técnicas e não técnicas.



Fonte: O Autor (2025).

A Figura 2 é uma representação minimalista de um SEP completo. Nela constam os três grandes subsistemas (de geração, transmissão e distribuição) do modo mais simples, facilitando a visão do fluxo de energia por todo o sistema. Os consumidores são conectados à rede elétrica através de um medidor de energia que irá determinar quanto eles consumiram de energia durante determinado intervalo de tempo.

Figura 2 – Fluxo de potência no SEP.



Fonte: O Autor (2025).

Em que:

- a) E_G : Energia total gerada;
- b) E_{ST} : Energia que flui pelo sistema transmissão;
- c) P_{ST} : Perdas técnicas no sistema de transmissão;
- d) E_{SD} : Energia que flui pelo sistema de distribuição;
- e) P_{SD} : Perdas técnicas no sistema de distribuição;
- f) E_{TC} : Energia total entregue ao consumidor;
- g) E_{NF} : Energia não faturada pela concessionária;
- h) E_F : Energia faturada pela concessionária.

Assim, a perda total de uma concessionária de energia pode ser definida pela diferença entre a energia comprada menos a energia faturada por ela (1). Por sua vez, as PNT podem ser determinadas por (2), ou seja, pela diferença entre as perdas totais e as PT (Ramos, 2014; Agência Nacional de Energia Elétrica, 2024).

$$\text{Perda Total} = \text{Energia Comprada} - \text{Energia Faturada} \quad (1)$$

$$PNT = Perda Total - PT \quad (2)$$

Além do problema financeiro causado, segundo Gitura (2006), as conexões ilegais podem causar problemas de segurança que envolvem desde choques elétricos menores até casos mais graves de eletroplessão, ou seja, aquela descarga que pode ocasionar até a morte do indivíduo. A qualidade de energia elétrica (QEE) também é afetada pelas ligações clandestinas, visto que o fluxo de potência que irá fluir pelo sistema é diferente daquele esperado, o que pode acarretar em riscos de incêndio e até apagões generalizados. De acordo com Olaoluwa (2017), as fraudes envolvendo energia elétrica podem gerar qualquer problema relacionado à QEE, como flutuações de tensão, sobretensões, subtensões, harmônicos, entre outras questões. É possível que um cliente que necessita de uma tensão maior (como um cliente industrial ou comercial) realize uma conexão ilegal em um nível de tensão mais baixo. Isso resulta em uma elevação da intensidade da corrente elétrica na rede, o que por sua vez, também acarreta no aumento das PT (Ferreira, 2008).

1.1 ABORDAGEM DO PROBLEMA E JUSTIFICATIVA

É fundamental identificar de forma correta a ocorrência de PNT em sistemas elétricos, visando reduzir seu montante e, conseqüentemente, os elevados custos financeiros e outros problemas a elas associados.

Atualmente, os modelos de Aprendizado de Máquina (AM) são amplamente explorados como a abordagem principal para lidar com a detecção de PNT. No entanto, alguns desses estudos apresentam resultados de classificação relativamente baixos ou utilizam um número limitado de modelos matemáticos para simular diferentes tipos de fraudes que podem ocorrer em infraestruturas avançadas de medição (*Advanced Metering Infrastructure* - AMI).

Neste estudo, foram implementados modelos de AM para determinar se um consumidor cometeu fraude no consumo de energia elétrica, visando melhorar o desempenho de detecção encontrado em outros estudos na literatura. Foram utilizados dois tipos de redes neurais para a classificação: uma rede *feedforward* do tipo *Multilayer Perceptron* (MLP) e uma rede recorrente do tipo *Long Short-Term Memory* (LSTM). Utilizando essas arquiteturas de rede, foram desenvolvidos diversos modelos, variando a taxa de amostragem dos dados de entrada e as variáveis de entrada. O modelo final proposto neste estudo possui dois ramos em paralelo, cada um contendo um desses tipos de redes para processar diferentes tipos de informações. Todos

os modelos foram comparados para determinar qual apresenta melhor precisão na tarefa de detecção.

1.2 OBJETIVOS DA PESQUISA

Estudar a detecção de PNT em sistemas de distribuição de energia elétrica, que têm como base medidores inteligentes de energia presentes em AMI, com o intuito de auxiliar na tomada de decisão por parte das concessionárias de energia elétrica.

Propor um modelo de AM que consiga detectar com exatidão se determinado consumidor cometeu fraude no consumo de energia elétrica ou não.

1.2.1 Objetivos Específicos

- a) Apresentar uma revisão bibliográfica sobre as PNT no Brasil e no mundo, e métodos utilizados para sua detecção;
- b) Realizar a correção de dados inconsistentes na base, para melhorar a qualidade dos dados que serão fornecidos aos modelos de AM;
- c) Estudar diferentes atributos que podem ser extraídos de séries temporais com o intuito de melhorar o desempenho de classificação dos modelos desenvolvidos;
- d) Propor novos modelos que irão realizar a detecção das amostras fornecidas, ou seja, detectar se dado consumidor cometeu fraude ou se seu perfil é regular. Esses modelos são aperfeiçoamentos dos modelos que já foram propostos em Souza *et al.* (2024).

1.3 ORGANIZAÇÃO DO TEXTO

O Capítulo 1 traz uma breve perspectiva sobre as PT e PNT e como essas podem afetar o SEP.

O Capítulo 2 aborda uma revisão da literatura sobre a PNT no mundo, no Brasil e o impacto financeiro que ela causa, além de mostrar trabalhos relevantes na literatura que tratam do tema da detecção desse tipo de perda.

No Capítulo 3 é apresentada a abordagem teórica sobre os métodos de AM utilizados para realizar a detecção de um consumidor que agiu de forma fraudulenta.

O Capítulo 4 aborda detalhadamente o problema a ser resolvido. Nele, são apresentadas a base de dados utilizada no estudo, os procedimentos de pré-processamento aplicados, os mo-

delos de fraude de energia elétrica empregados. Também é apresentada uma breve explicação sobre a técnica *Uniform Manifold Approximation and Projection* (UMAP), os atributos extraídos da base, os modelos desenvolvidos e a métrica utilizada para avaliar seu desempenho.

No Capítulo 5 encontram-se as informações sobre as simulações realizadas e os resultados obtidos.

Por fim, o Capítulo 6 traz as conclusões do trabalho, além de sugestões para sua continuidade.

2 REVISÃO DA LITERATURA

A PNT é um problema mundial que afeta tanto países subdesenvolvidos quanto em desenvolvimento, e até mesmo os desenvolvidos. Um relatório publicado em 2017 indicou que, naquele ano, foram perdidos em torno US\$ 96 bilhões em todo o mundo (Northeast Group LLC, 2017). Em seu estudo mais recente, realizado em 2021 com dados de 138 países, a Northeast Group LLC (2021) concluiu que esse valor chegou a US\$ 101,2 bilhões. Nos Estados Unidos, em 2009, estimou-se que as concessionárias tiveram um prejuízo anual de cerca de US\$ 6 bilhões (McDaniel; McLaughlin, 2009). A BC Hydro, principal concessionária da província da Colúmbia Britânica no Canadá, relatou que o prejuízo anual por fraudes relacionadas à energia elétrica é de aproximadamente CA\$ 100 milhões, o que equivale à energia necessária para abastecer 77.000 residências durante um ano (BC Hydro, 2011). Entre 2010 e 2018, em 35 países da União Europeia, os valores combinados das PTs e PNTs nos sistemas de distribuição variaram de 2% a 14% em relação à energia injetada (Council of European Energy Regulators, 2020). Em países subdesenvolvidos e em desenvolvimento, a situação tende a ser ainda pior. Os principais países emergentes registram perdas anuais de US\$ 58,7 bilhões devido às PNTs (Savian *et al.*, 2021).

As perdas financeiras são críticas para as concessionárias de energia, pois além de impedir que consigam realizar investimentos visando a melhoria da qualidade do serviço, elas acabam tendo dificuldades de manter os funcionários na empresa com o corte no orçamento, e em muitos casos essas empresas estão próximas à falência (Smith, 2004). O relatório técnico de Kaufmann, Kraay e Zoido-Lobaton (2002) para o Banco Mundial desenvolveu indicadores para avaliar a governança de um local, e, segundo Smith (2004), o nível de governança de um país tem correlação com as PNT. De acordo com este autor, um país que possui alta instabilidade política, elevado nível de violência, corrupção por parte da população e dos governantes, falta de transparência nas atividades do Estado e falta de imposição das leis, tende a ter taxas mais elevadas de PNT.

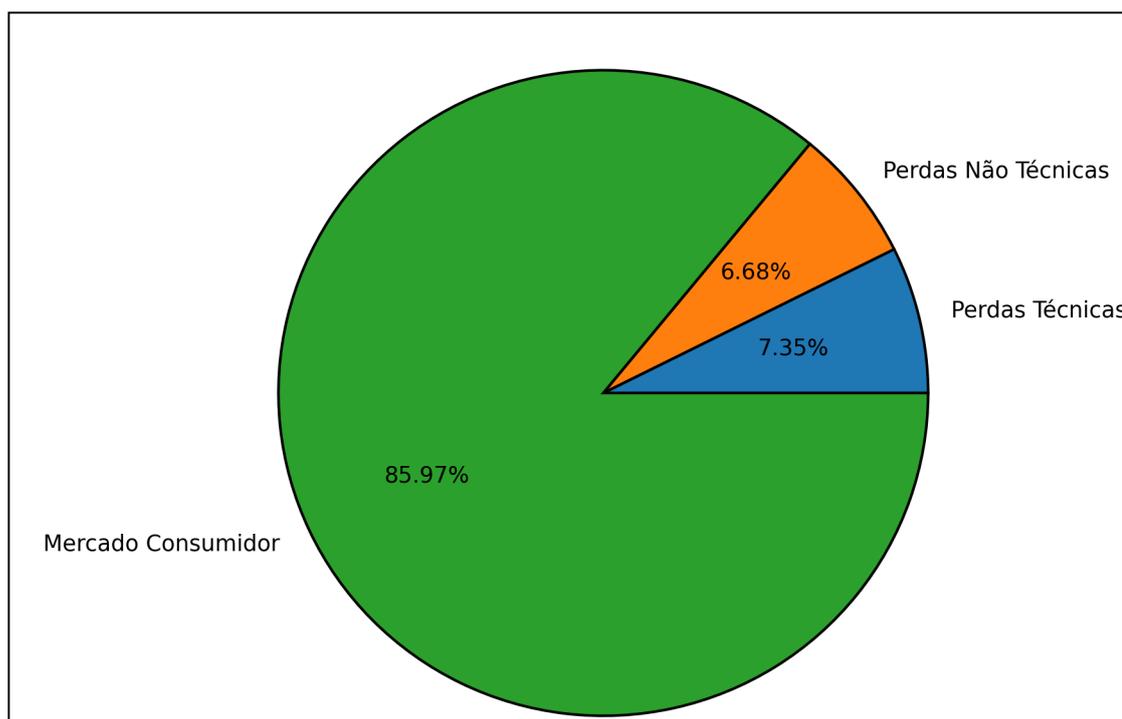
A seguir, apresenta-se o estudo de PNT no Brasil, baseado em relatórios anuais da Agência Nacional de Energia Elétrica (ANEEL), e é demonstrado como ela se mantém presente em todos os Estados da Federação e os prejuízos causados. Também são expostos trabalhos voltados à detecção desse tipo de perda.

2.1 PNT NO BRASIL

As PNT prejudicam não apenas a concessionária de energia, mas toda a sociedade. No momento da revisão tarifária pela distribuidora, é o consumidor honesto que acaba arcando parcialmente pela prática fraudulenta de outros. Ou seja, o repasse das perdas comerciais é, até certo limite definido pela ANEEL, repassado aos consumidores na forma de aumento da tarifa média de energia elétrica (TME). De acordo com a Agência Nacional de Energia Elétrica (2024), os montantes das PNT são divididos pelo mercado de baixa tensão faturado, visto que é nesse mercado que os valores dessas perdas são mais elevados.

A Figura 3 contém os percentuais de cada perda em relação ao montante de energia injetada na rede de distribuição no Brasil em 2023. Tanto as PT quanto as PNT são problemas recorrentes na realidade brasileira. Dados fornecidos pela ANEEL em seu portal de relatórios (Agência Nacional de Energia Elétrica, 2025) indicam que, entre 2008 e 2023, essas perdas giraram em torno de 7,31% e 6,59%, respectivamente, ao ano sobre a energia injetada.

Figura 3 – Percentuais de perdas em relação a energia injetada na rede de distribuição no Brasil em 2023.



Fonte: Agência Nacional de Energia Elétrica (2025).

Vale ressaltar que destoa o valor de PNT por região no Brasil. Em relação à energia injetada, em 2023, a região Norte apresenta a maior taxa, com 19,75%, e a região Sul tem a menor, com 2,94%. A Tabela 1 contém os valores percentuais de PNT de todas as regiões do país.

Tabela 1 – Percentuais de PNTs sobre a energia injetada por região em 2023.

Região				
Norte	Sudeste	Nordeste	Centro Oeste	Sul
19,75%	6,92%	5,63%	4,24%	2,94%

Fonte: Agência Nacional de Energia Elétrica (2025).

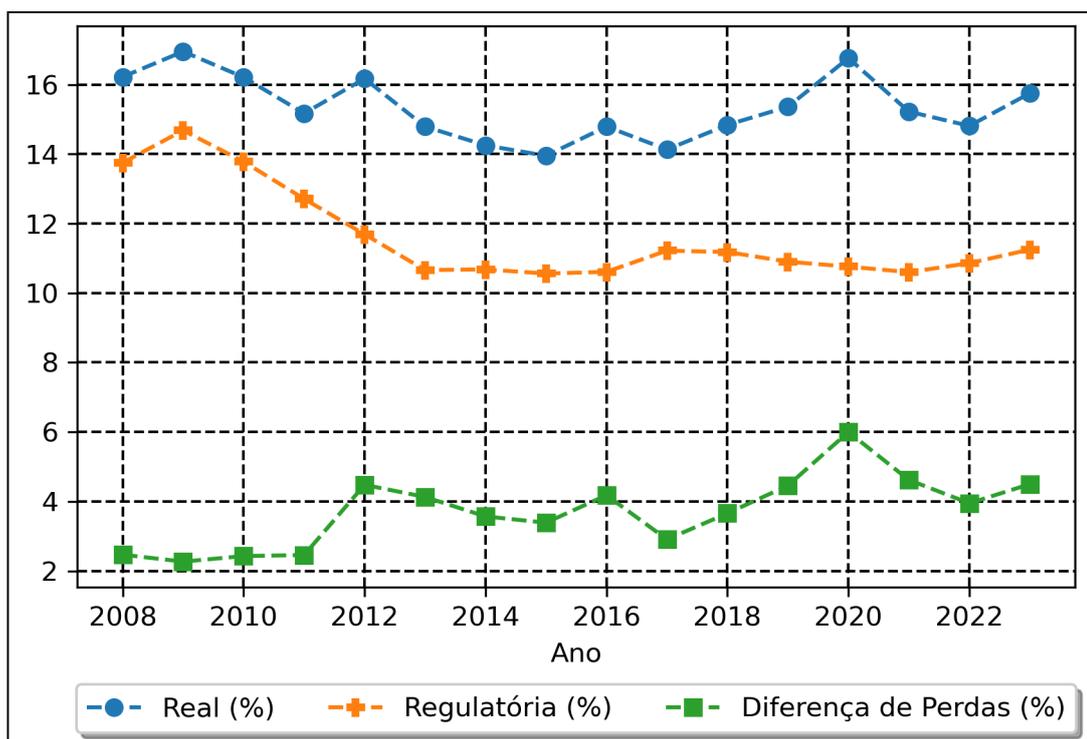
Concessionárias de grande porte (aquelas com mercado maior que 1 TWh e que atendem mais de 500 mil unidades consumidoras ou que possuem mais de 15 mil km de rede elétrica) são responsáveis por praticamente toda a PNT no Brasil, devido ao tamanho do seu mercado e à dificuldade de combater o problema. Ainda, segundo a Agência Nacional de Energia Elétrica (2024), os níveis de PNT dependem da administração por parte das concessionárias e de características socioeconômicas em cada área de concessão.

2.1.1 PNT Real e Regulatória

Deve-se frisar a diferença entre perda não técnica real (PNTR) e perda não técnica regulatória (PNTReg). A PNTR é aquela que de fato ocorre no sistema, enquanto que a PNTReg é a que é reconhecida na tarifa de energia. Os valores de PNTReg são calculados levando-se em conta o desempenho das concessionárias e fatores socioeconômicos da área de concessão. Este trabalho não trata sobre o cálculo desse tipo de PNT, mas maiores detalhes podem ser encontrados no submódulo 2.6 do módulo 2 do Proret (Agência Nacional de Energia Elétrica, 2015). Caso o valor de PNTR seja maior que o valor regulatório, a diferença de custos entre esses valores é de responsabilidade da concessionária. Sendo assim, as empresas de energia devem sempre atuar no sentido de reduzir esse tipo de perda, para amenizar o seu prejuízo.

Como já mencionado, é no mercado de baixa tensão que esses valores são mais elevados, devido à complexidade de combater esse tipo de perda no sistema de distribuição de energia elétrica. Na Figura 4 pode ser observado o valor médio ponderado das PNTR, PNTReg e da diferença entre elas, no Brasil, entre 2008 e 2023. Essa média ponderada é realizada para considerar concessionárias que possuem mais representatividade em termos de perdas.

Figura 4 – PNTR e PNTReg nos últimos anos no Brasil sobre o mercado de baixa tensão.



Fonte: Agência Nacional de Energia Elétrica (2025).

Na Tabela 2 encontram-se os valores médios de PNTR por região obtidos no mesmo período. Observa-se que a região Norte contabiliza mais da metade de toda PNT do país.

Tabela 2 – Percentuais de PNTR por região sobre o mercado de baixa tensão.

Região				
Norte	Sudeste	Nordeste	Centro Oeste	Sul
52,14%	15,30%	12,76%	8,29%	8,06%

Fonte: Agência Nacional de Energia Elétrica (2025).

De acordo com dados da ANEEL, Tabela 3, no Estado de Pernambuco, entre os anos de 2013 e 2024, o valor da TME praticamente dobrou, e o valor de PNTR apresentou uma tendência de crescimento no período.

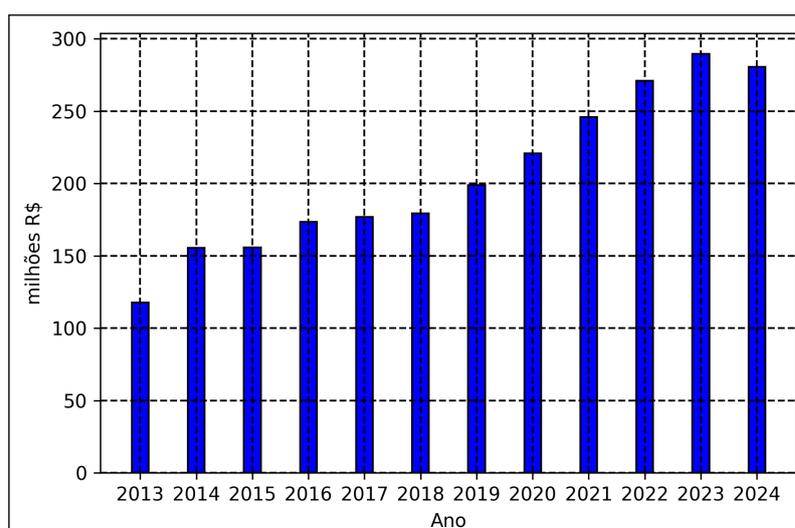
Tabela 3 – Valores da TME e de PNTR no Estado de Pernambuco.

Ano	TME (R\$/MWh)	PNTR (MWh)
2013	132,43	887.766
2014	158,25	982.433
2015	150,80	1.033.313
2016	169,15	1.025.039
2017	170,40	1.038.131
2018	177,62	1.009.059
2019	194,19	1.023.384
2020	211,98	1.042.392
2021	235,46	1.044.066
2022	257,39	1.053.053
2023	279,12	1.037.098
2024	257,00	1.091.630

Fonte: Agência Nacional de Energia Elétrica (2025).

Mesmo com a redução da TME em 2024, os custos associados a ela continuam elevados visto que a PNTR continua aumentando. Assim, o prejuízo financeiro causado para a concessionária de energia em Pernambuco vem aumentando ano após ano, com ligeiro decréscimo em 2024 e pode ser visualizado na Figura 5, totalizando R\$ 2,46 bilhões no período.

Figura 5 – Custo da PNTR no Estado de Pernambuco.

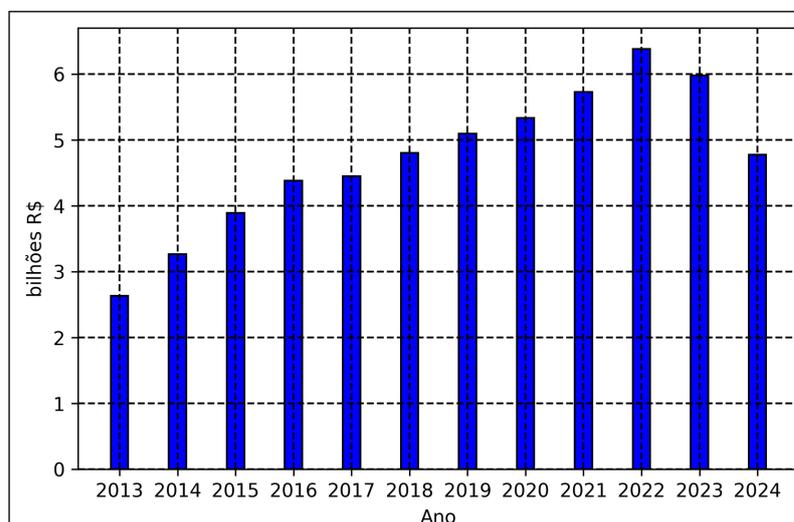


Fonte: Agência Nacional de Energia Elétrica (2025).

Da mesma forma, no Brasil, o custo da PNTR aumenta ano após ano, como evidenciado na Figura 6, resultando em um déficit total de R\$ 56,73 bilhões para as concessionárias de

energia. No entanto, esse valor deve ser ainda maior, pois a redução dos custos observada em 2023 e 2024 em relação aos anos anteriores se deve ao fato de que nem todas as concessionárias forneceram informações sobre as perdas à ANEEL nesses anos.

Figura 6 – Custo da PNTR no Brasil.



Fonte: Agência Nacional de Energia Elétrica (2025).

Sendo assim, esse é um tema de grande relevância, e no mundo inteiro vem sendo realizadas pesquisas com diferentes abordagens que auxiliem na redução ou solução desse problema.

2.2 TRABALHOS RELEVANTES

As primeiras tentativas de detecção de PNT baseavam-se em inspeções de campo conduzidas por funcionários das concessionárias. Isso exigia uma grande quantidade de recursos humanos, resultava em custos operacionais elevados (Nizar *et al.*, 2006), e também era considerado ineficiente (Nagi *et al.*, 2010; Saeed *et al.*, 2020).

Na literatura existem diversos trabalhos que abordam de diferentes formas o problema de PNT. Zheng *et al.* (2018) separam os métodos de solução em duas categorias: soluções baseadas em *hardware* e soluções baseadas em dados. Viegas *et al.* (2017) ainda afirmam que existe uma terceira abordagem. Estes autores a denominam de solução teórica e que aborda fatores demográficos e socioeconômicos que possam auxiliar na tomada de decisões por parte dos operadores do sistema elétrico para prevenir o fenômeno de PNT.

As soluções baseadas em *hardware* tratam de infraestruturas e *designs* de dispositivos de medição específicos. Por exemplo, podem ser instalados circuitos integrados (CI) que detectam adulteração nos medidores (Ngamchuen; Pirak, 2013). O CI faz medições de potência, energia e outras grandezas fundamentais, e, quando detecta alguma anormalidade, envia um sinal de

interrupção para um microcontrolador que irá realizar uma rotina de verificação e classificar o tipo de adulteração que foi cometida. Esse trabalho teve uma falsa taxa de detecção de apenas 2,13%. No estudo de Dey *et al.* (2010), é apresentado um dispositivo que utiliza dois transformadores de corrente (TCs) para auxiliar na identificação de fraudes, como em situações em que o condutor de neutro é desconectado ou ocorre adulteração por meio de um ímã magnético. Um TC mede a corrente da fase e o outro a corrente do neutro, com esses dados sendo enviados à entrada AC-DC de um microcontrolador. Caso haja alguma discrepância entre os valores das correntes, o microcontrolador detecta a fraude. Porém, os maiores problemas relacionados à abordagem via *hardware* são os custos de fabricação, instalação e manutenção dos novos medidores.

A detecção baseada em dados (também conhecida como baseada em não *hardware*) tem recebido bastante atenção, e é a abordagem que mais vem sendo utilizada para a detecção de PNT (Viegas *et al.*, 2017). Esse método trata de analisar dados de consumidores, pois se assume que as PNTs geram desvios nos padrões de consumo de energia elétrica e de outras características dos consumidores.

Um dos primeiros trabalhos baseados em dados que obteve relativo sucesso no combate à PNT na prática foi conduzido por Nagi *et al.* (2010). Utilizando uma base de dados rotulada, na qual as amostras já estavam previamente marcadas indicando se o consumidor havia cometido fraude ou não, foi treinada uma Máquina de Vetores de Suporte (*Support Vector Machine - SVM*) para realizar a classificação. Esse sucesso foi relativo, pois a taxa de detecção de fraude nesse trabalho foi de apenas 60%. Anteriormente, na Malásia (onde o estudo foi conduzido para auxiliar uma empresa de energia local), a taxa de detecção era de apenas 3%, pois o método empregado até então era de enviar equipes aos possíveis locais para realizar a inspeção.

Com o surgimento das redes elétricas inteligentes (*Smart Grids - SG*) e a implementação das AMI, uma vasta quantidade de dados está sendo armazenada e empregada para análise por concessionárias de energia. A AMI é um sistema composto por medidores de energia elétrica inteligentes equipados com portas de comunicação, redes de comunicação e outros dispositivos como concentradores e gerenciadores de dados. Mediante esses medidores inteligentes, os dados são coletados e enviados para uma central de operações. Embora as AMI possam auxiliar na detecção de fraudes relacionadas à energia elétrica, devido à sua capacidade de monitoramento, a introdução de canais de comunicação também pode abrir brechas para novas formas de fraude, como ciberataques que manipulam os dados coletados e/ou enviados (Wang; Lu, 2013).

No estudo realizado por Jokar, Arianpoo e Leung (2016), foi apresentado um esquema de

detecção em uma AMI para uma área alimentada por um transformador. Nesse esquema, é feita a comparação do consumo de energia do medidor do transformador com o consumo total de todos os medidores inteligentes presentes naquela área/vizinhança. Caso ocorra uma diferença, os consumidores alimentados pelo transformador serão marcados como suspeitos de terem cometido fraude. Foram propostas equações matemáticas, baseadas em dados de consumo real, para modelar seis diferentes tipos de fraudes que podem ocorrer através da AMI. Por fim, é treinada uma SVM com os dados obtidos pelo medidor inteligente e as amostras geradas pelos modelos matemáticos, a fim de classificar se houve alguma fraude cometida por um consumidor ou se há algum problema no medidor.

Mais recentemente, trabalhos envolvendo modelos de Aprendizagem Profunda (*Deep Learning* - DL) vêm sendo empregados para realizar a detecção de consumidores desonestos. Um modelo híbrido composto por um MLP e uma Rede Neural Convolucional (*Convolutional Neural Network* - CNN) é proposto em Zheng *et al.* (2018). São utilizados dados de consumo de energia diários que já estão rotulados. O MLP tem o objetivo de realizar o aprendizado global desses dados diários em 1-D, ou seja, como eles oscilam dia após dia. Porém, como esses dados oscilam de maneira independente, o aprendizado se torna mais difícil utilizando apenas o MLP. Por isso, os autores propõem acrescentar uma CNN que tem o objetivo de processar os dados em um formato 2-D, ou seja, os dados são convertidos de diários para semanais, e assim alimentarem a CNN. Por fim, a saída de cada rede é somada de formada ponderada e o resultado é introduzido em uma função de ativação sigmóide para determinar a probabilidade de um consumidor ter cometido fraude. Para aprimorar a eficácia do modelo, foram executadas diversas etapas de pré-processamento, incluindo a aplicação de interpolação linear para preencher lacunas nos dados e a utilização da regra empírica, com dois desvios padrão da média (regra dos 95%), para identificar valores discrepantes (*outliers*) no conjunto de dados.

Zanetti *et al.* (2019) propuseram duas novas equações para a geração de amostras maliciosas, complementando as já sugeridas por Jokar, Arianpoo e Leung (2016), e utilizaram uma abordagem semelhante que também requer dados do medidor do transformador de baixa tensão. O método inclui um "detector" que compara os relatórios do medidor do transformador e os medidores inteligentes residenciais. Posteriormente, uma abordagem baseada em máquina de estados, que define três estados na rede, normal (G^1), suspeito (G^2) ou anormal (G^3), é empregada para identificar usuários nessas três diferentes categorias.

Uma nova equação matemática foi proposta por Messinis, Rigas e Hatzargyriou (2019) para simular um tipo de fraude em um medidor inteligente. Ela considera que o ataque no

medidor aumenta linearmente ao longo do tempo; ou seja, o consumo de energia diminui de forma linear em vez de diminuir abruptamente, como normalmente acontece. Em seguida, com base em algumas equações de Jokar, Arianpoo e Leung (2016) e nesta nova equação proposta, é utilizado um modelo híbrido que emprega análise de sensibilidade de tensão, otimização de sistemas de energia e uma SVM para realizar a detecção de PNT.

No caso das bases de dados sobre PNTs, há o problema do desbalanceamento de classes quando elas já são rótuladas, pois, na prática, há uma quantidade significativamente maior de dados de consumidores honestos em comparação aos desonestos, o que pode acarretar dificuldades de aprendizado para os algoritmos de AM (Domingues *et al.*, 2018).

Usando o mesmo conjunto de dados de Zheng *et al.* (2018) (que é uma base desbalanceada), os autores de Khan *et al.* (2020) utilizaram uma arquitetura denominada VGG-16, que consiste de uma rede CNN composta por uma combinação de cinco camadas convolucionais e de *pooling*, além de mais três camadas totalmente conectadas para extrair automaticamente características dos dados. Por fim, foi empregado o *Extreme Gradient Boosting* (XGBoost), que é um algoritmo baseado em árvores de decisão para realizar a classificação final. Os hiperparâmetros do XGBoost foram otimizados utilizando a metaheurística de otimização baseada no comportamento dos vaga-lumes (*Firefly Algorithm* - FA). Para lidar com o problema do conjunto de dados desbalanceado, a técnica *Adaptive Synthetic Sampling* (ADASYN) foi utilizada para fazer a sobreamostragem da classe minoritária.

Um *ensemble* formado por vários métodos de *boosting* e *bagging* foi desenvolvido para detectar PNTs (Gunturi; Sarkar, 2021). A ideia por trás do *ensemble* é treinar diversos modelos e combinar os resultados de cada um em um único modelo que demonstre melhor desempenho preditivo do que cada modelo individualmente. Utilizando as mesmas equações descritas em Jokar, Arianpoo e Leung (2016), foram geradas amostras maliciosas, porém ao invés de criar um problema multiclasse, o problema foi formulado como uma classificação binária. A eficácia dos modelos foi avaliada com o uso de técnicas de subamostragem (*near miss*) e sobreamostragem (*Synthetic Minority Over-sampling Technique* - SMOTE), com esta última apresentando resultados ligeiramente superiores.

Em Chuwa e Wang (2021), todos os exemplos maliciosos gerados pelas equações dos trabalhos anteriores (Jokar; Arianpoo; Leung, 2016; Zanetti *et al.*, 2019; Messinis; Rigas; Hatziargyriou, 2019), além de equações adicionais, foram empregados para criar uma ampla gama de casos de roubo de energia. O estudo tratou o problema como um problema de classificação multiclasse, utilizando vários algoritmos de AM para sua resolução. Os modelos foram com-

parados para avaliar seu desempenho tanto em classificação global quanto por classe. Além disso, foram conduzidos testes nos quais os parâmetros das equações que geraram os exemplos maliciosos (como intensidade e duração dos ataques) foram ajustados para investigar o comportamento dos modelos. Esses resultados fornecem uma base sólida para futuras pesquisas no campo da detecção de roubo de energia, visto que utiliza a maior quantidade de diferentes possíveis fraudes e/ou defeitos que podem acontecer em medidores inteligentes de uma AMI.

No trabalho de Bastos *et al.* (2023), cinco classificadores foram treinados utilizando algoritmos desenvolvidos especificamente para a classificação de séries temporais. Os algoritmos empregados foram: *Time Series Forest* (TSF), *Residual Network* (ResNet), *Inception Time*, *Time Le-Net* (t-LeNet) e *Multi Channel Deep Convolutional Neural Network* (MCDCNN). Esses algoritmos foram implementados por meio da biblioteca Sktime (Löning *et al.*, 2019), que oferece uma ampla gama de métodos voltados para a classificação e previsão de séries temporais. Além disso, os autores propuseram a criação de um *ensemble* composto por esses cinco classificadores, visando melhorar a taxa de detecção das PNT. A decisão sobre a detecção de uma PNT pelo *ensemble* é tomada com base no voto da maioria (*hard voting*).

Menos comuns, mas ainda utilizado em alguns trabalhos, são os métodos não supervisionados. Eles possuem a inerente vantagem de não necessitar de dados rotulados para realizar a classificação. Dados de consumo de energia e de tensão de medidores inteligentes foram utilizados em um método de aprendizagem não supervisionada (Gao; Foggo; Yu, 2019). Os autores treinam um modelo de regressão linear que tem como entrada essas duas variáveis para tentar estimar o próprio consumo (previsão de carga). Nessa metodologia, todos os consumidores conectados ao secundário de um transformador são analisados ao mesmo tempo. Após o treinamento, o conjunto de teste é fornecido ao modelo, onde são determinados os resíduos (diferença entre o valor real e o previsto). É provado que o resíduo de um consumidor desonesto é negativo, enquanto que os resíduos dos outros consumidores conectados ao mesmo transformador é ligeiramente positivo. Por fim, é feita uma "pontuação de anomalia" sobre esses resíduos, a qual pode ser utilizada pelo operador do sistema na tomada de decisão. Quanto maior a pontuação, maior deve ser a prioridade sobre a investigação do consumidor.

Outros trabalhos exploram técnicas de otimização para selecionar ou extrair melhores atributos (*features*), visando reduzir a dimensionalidade do problema e, conseqüentemente, melhorar tanto o tempo de processamento quanto a precisão da detecção das PNT. Em Ramos *et al.* (2011), foram utilizadas duas bases de dados, uma de consumidores industriais e outra de comerciais. Porém, ao invés de utilizar apenas dados de consumo de energia elétrica, foram em-

pregadas outras grandezas elétricas, como potência do transformador (kVA), fator de potência, potência instalada (kW) e outras. Os classificadores Floresta de Caminhos Ótimos (*Optimum-Path Forest* - OPF), SVM, MLP, Mapa Auto-Organizável de Kohonen (*Self-Organizing Map* - SOM) e *k-Nearest Neighbors* (kNN) foram utilizados para realizar a classificação, e obtiveram resultados em torno de 60% de acurácia na detecção em ambas as bases. A técnica de Busca Harmônica (*Harmony Search* - HS) foi então utilizada para reduzir a dimensionalidade dos dados. Posteriormente, com o novo conjunto reduzido, o classificador OPF foi novamente utilizado, resultando em uma acurácia de 96,5% na base de consumidores industriais e 92,60% na base de consumidores comerciais.

Em uma continuação desse trabalho, outras técnicas, além da HS para seleção de atributos, são testadas e comparadas em um classificador OPF (Ramos, 2014). Destaca-se a técnica de Otimização por Enxame de Partículas (*Particle Swarm Optimization* - PSO), que apresentou a melhor precisão entre todas as avaliadas.

Na prática, existe certa dificuldade em encontrar bases de dados já rotulados que indiquem fraudes entre consumidores. Isso ocorre porque tais bases geralmente são propriedade privada de empresas de energia, e há também questões relacionadas à privacidade dos consumidores. Por essa razão, muitos estudos na literatura utilizam bases de dados sem rótulos, recorrendo a modelos matemáticos para criar amostras que identifiquem fraudes ou falhas nos medidores. Uma vantagem dessa abordagem é a capacidade de resolver automaticamente o problema do desbalanceamento dos dados ao gerar amostras equivalentes para cada tipo de fraude, equiparando-se ao número de amostras de consumidores honestos.

Este trabalho aborda o problema da detecção de PNT usando dados de consumidores tidos como honestos, enquanto as fraudes ou defeitos nos medidores são simulados por equações matemáticas. Para isso, foram empregadas duas arquiteturas distintas de redes neurais para identificar consumidores fraudulentos. O próximo capítulo discute a teoria fundamental por trás desses métodos.

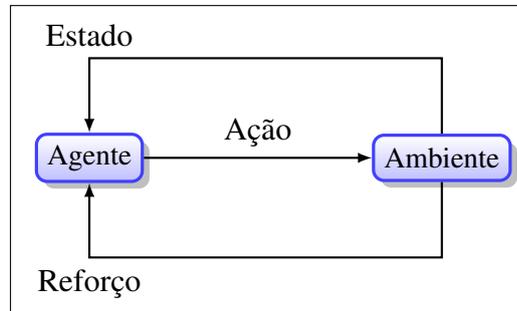
3 APRENDIZAGEM DE MÁQUINA

AM é um tipo de Inteligência Artificial (IA) que utiliza modelos matemáticos treinados com um conjunto de dados para resolver determinado problema. Ou seja, um modelo de AM adquire seu próprio conhecimento, extraindo padrões de dados (Goodfellow; Bengio; Courville, 2016). Existem três principais paradigmas de AM (Raschka; Mirjalili, 2019), que são:

- a) **Aprendizagem Supervisionada:** Neste tipo de AM, o objetivo é produzir um modelo (função matemática) que melhor estime o valor da saída a partir de dados de entrada, para posteriormente fazer previsões em dados nunca vistos (Müller; Guido, 2016). Inicialmente, um conjunto de dados (conjunto de treinamento) em que cada amostra contém um par de entradas (atributos/*features*) e saídas (rótulos/*labels*) conhecido é apresentado ao modelo para que ele seja treinado. Finalizado o treinamento, um conjunto de testes com dados nunca vistos pode ser aplicado ao modelo para fazer previsões e verificar o desempenho do modelo. A aprendizagem supervisionada se subdivide em problemas de classificação (aqueles em que se deseja prever um valor discreto) ou de regressão (em que o objetivo é prever um valor contínuo). A SVM e o MLP são algoritmos clássicos que se encaixam nesse tipo de aprendizado;
- b) **Aprendizagem Não Supervisionada:** Na Aprendizagem Não Supervisionada, lida-se com dados que não estão rotulados ou que sua estrutura não é conhecida, ou seja, é um tipo de aprendizagem em que se utiliza apenas dados de entrada para o treinamento (Burkov, 2019). Então, através de técnicas não supervisionadas, explora-se a estrutura desconhecida dos dados para tentar extrair alguma informação relevante. Os métodos de *clusterização*, como o *k-means* e o agrupamento hierárquico são exemplos de técnicas não supervisionadas;
- c) **Aprendizagem por Reforço:** Os modelos de Aprendizagem por Reforço, muito utilizados na área de controle e robótica, são treinados para tomar uma sequência de decisões. Através de um processo de tentativa e erro, o algoritmo tenta encontrar a melhor solução para um problema. Para isso, o algoritmo é reforçado positivamente quando se trata de uma decisão correta ou negativamente no caso de uma incorreta (recompensas ou penalidades). De acordo com Raschka e Mirjalili (2019), o objetivo final é maximizar a recompensa total. Na Figura 7 apresenta-se o diagrama básico desse tipo de aprendizagem. O algoritmo (agente) efetua alguma ação no ambiente, que é interpretada gerando

um reforço (positivo ou negativo) e modificando o estado do mesmo. Como exemplos desse tipo de aprendizado, citam-se os algoritmos *TD learning* e *Q-learning*.

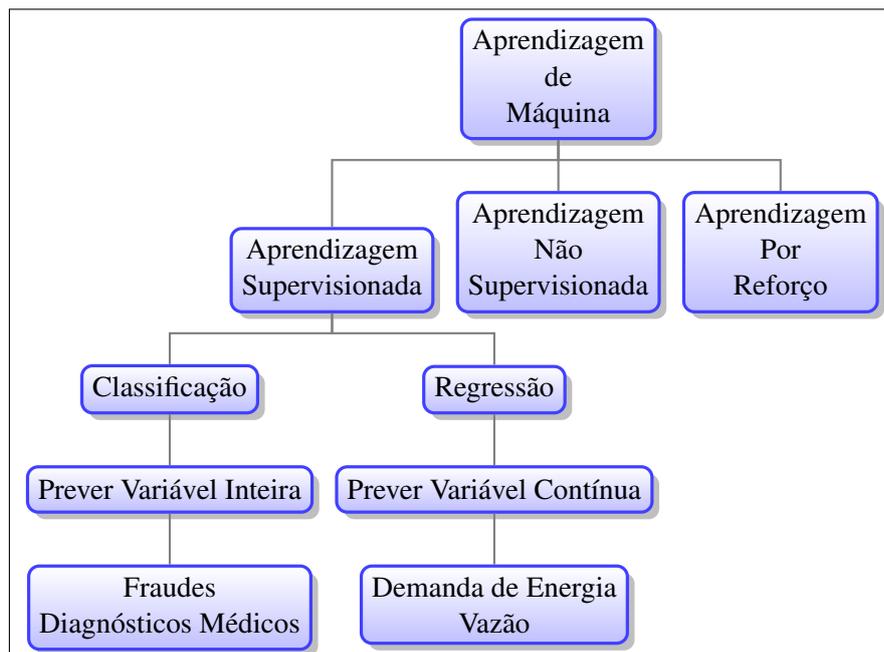
Figura 7 – Aprendizagem por Reforço.



Fonte: O Autor (2025).

Pode ser observado na Figura 8 os tipos de AM e exemplos de problemas que podem ser resolvidos nos casos da aprendizagem supervisionada.

Figura 8 – Tipos de Aprendizagem de Máquina.



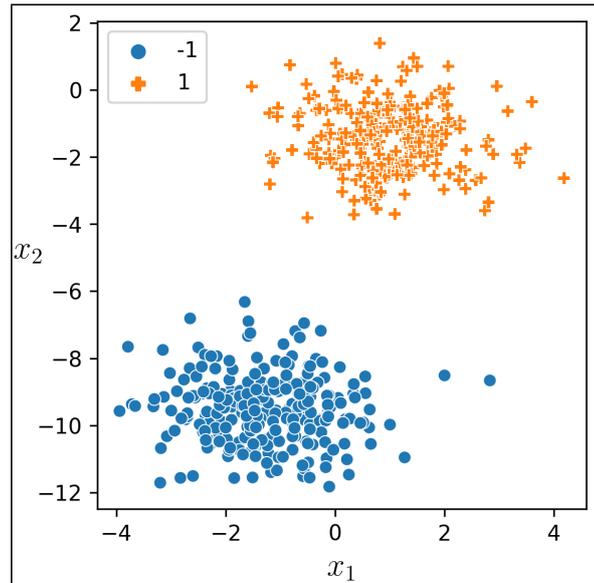
Fonte: O Autor (2025).

Neste trabalho, apenas métodos supervisionados serão empregados. Como já foi mencionado, os métodos supervisionados são utilizados para resolver problemas de classificação e de regressão:

- a) **Classificação:** A Figura 9 é de um problema linearmente separável de classificação binário, que possui duas variáveis (atributos), x_1 e x_2 , e cada amostra pode assumir apenas uma das duas classes. Já na Figura 10, pode novamente ser observado um problema

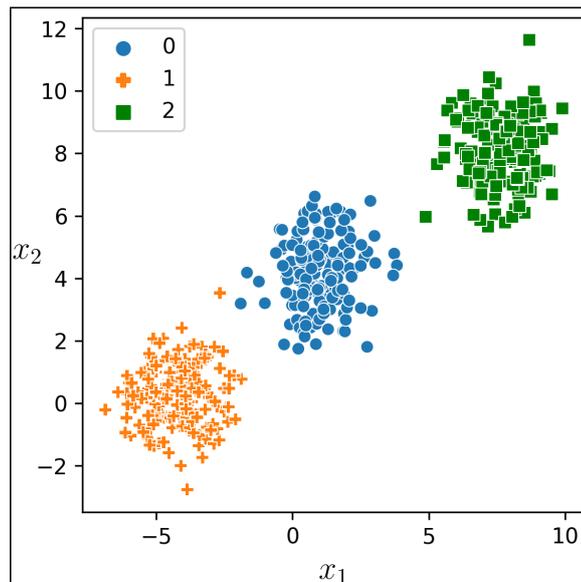
com dois atributos, porém que possui mais de duas classes (problema multiclass). A detecção de PNTs pode ser formulada como um problema de classificação binária que determina apenas se houve fraude ou não, ou como um problema de multiclass que determina se houve fraude (e identifica o tipo). Em ambos os casos, devido à alta dimensionalidade dos dados de consumo de energia, eles não são linearmente separáveis.

Figura 9 – Problema de classificação binário.



Fonte: O Autor (2025).

Figura 10 – Problema de classificação multiclass.

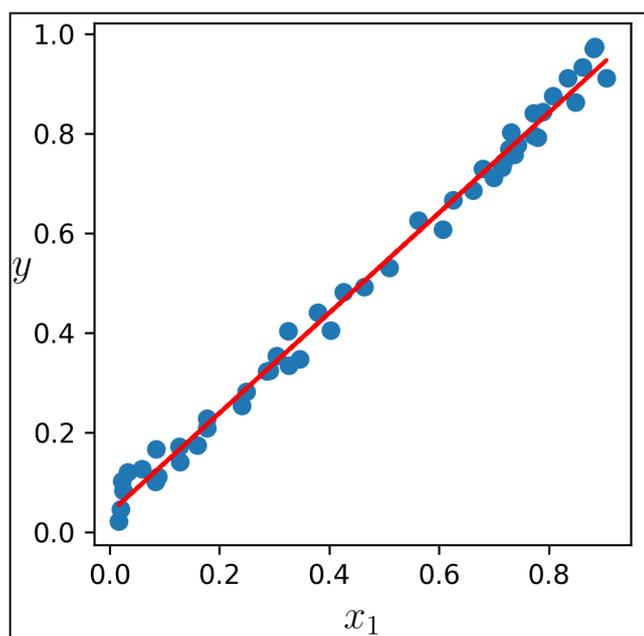


Fonte: O Autor (2025).

- b) Regressão: O problema de regressão é prever um valor contínuo de saída, dado um vetor de variáveis de entrada x (Bishop, 2006). Observa-se na Figura 11 uma regressão linear.

Por questão de visualização, considerou-se apenas uma variável de entrada x_1 e a variável de saída y . O objetivo é ajustar uma linha reta aos dados que minimize a distância entre esses e a respectiva linha. Após isso, com os valores obtidos dos coeficientes angular e linear, pode-se utilizar novos dados para fazer uma predição de um valor real.

Figura 11 – Problema de regressão.



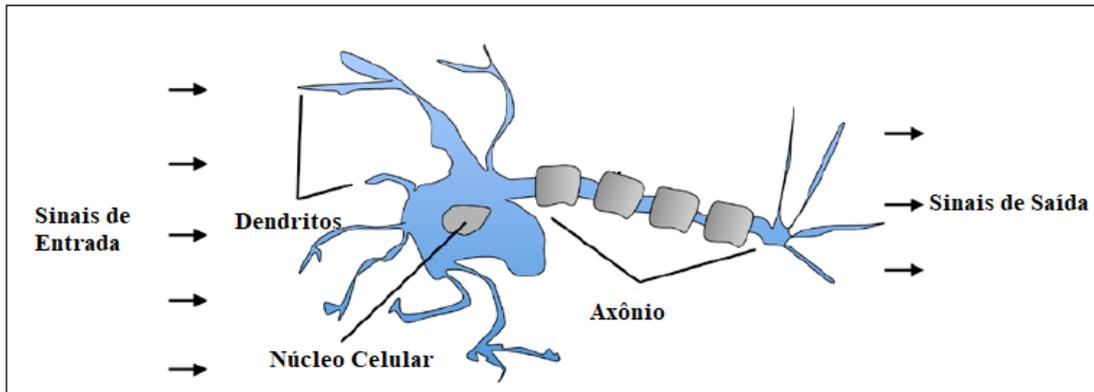
Fonte: O Autor (2025).

A seguir será apresentada a fundamentação teórica das duas arquiteturas de AM utilizadas neste trabalho para solucionar o problema de detecção de PNTs: MLP e LSTM.

3.1 PERCEPTRON MULTICAMADAS - MLP

O MLP é um dos modelos de Redes Neurais Artificiais (RNA) mais conhecidos no campo da AM supervisionada. O clássico trabalho de McCulloch e Pitts (1943) introduziu o modelo matemático de um neurônio artificial. Esse é comparado ao neurônio biológico do ser humano, que tem a função de processar e transmitir sinais elétricos, através dos dendritos, do núcleo celular e do axônio. Na Figura 12 pode ser observado cada elemento do neurônio biológico, em que os dendritos têm a função de receber estímulos (sinais de entrada); o núcleo celular processa esses sinais e posteriormente eles são transmitidos para outras células através do axônio.

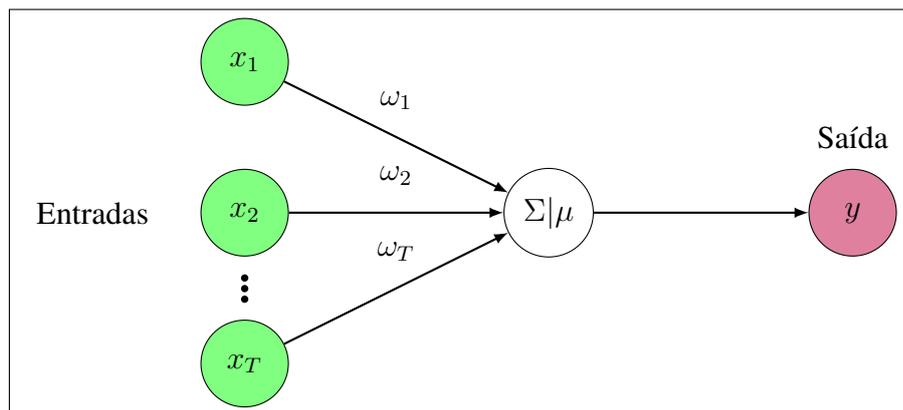
Figura 12 – Neurônio biológico.



Fonte: Adaptado de Raschka e Mirjalili (2019).

Assim, o neurônio artificial definido por McCulloch e Pitts (1943) era simplesmente uma porta lógica com diversas entradas e uma única saída binária. Múltiplos sinais chegavam no dendrito e eram somados no corpo celular. Caso esse resultado estivesse acima de um limite (θ) estabelecido, um sinal de saída era gerado e transmitido através do axônio (Raschka; Mirjalili, 2019). A Figura 13 e a Equação (3) retratam esse neurônio e o seu modelo matemático, em que \mathbf{x} e \mathbf{w} representam os vetores de entradas e de pesos sinápticos, respectivamente. Porém, esse trabalho tratou apenas de definir um modelo para o neurônio biológico (modelo MCP), ou seja, ele não tratava sobre técnicas de aprendizagem.

Figura 13 – Neurônio MCP.



Fonte: O Autor (2025).

$$z = \mathbf{w} \cdot \mathbf{x} = \sum_{t=1}^T w_t x_t \quad (3)$$

Então, funções de ativação, μ , podem ser aplicadas em z para obter determinado resultado (informação) na saída sobre o conjunto de dados utilizados. A Tabela 4 contém algumas

funções clássicas encontradas na literatura (Braga; Ludermir; Carvalho, 2007; Haykin, 2009; Goodfellow; Bengio; Courville, 2016)

Tabela 4 – Funções de ativação comumente utilizadas na literatura.

Funções de Ativação	
Função limiar (degrau)	$y = \mu(z) = \begin{cases} 1, & \text{se } z \geq \theta \\ 0, & \text{se } z < \theta \end{cases}$
Função sinal	$y = \mu(z) = \begin{cases} 1, & \text{se } z \geq \theta \\ -1, & \text{se } z < \theta \end{cases}$
Função linear	$y = \mu(z) = Bz$
Sigmoidal	$y = \mu(z) = \frac{1}{1+e^{(-\Gamma z)}}$
Tangente Hiperbólica	$y = \mu(z) = \frac{e^{(2\Gamma z)} - 1}{e^{(2\Gamma z)} + 1}$
ReLU	$y = \mu(z) = \begin{cases} Bz, & \text{se } z \geq \theta \\ 0, & \text{se } z < 0 \end{cases}$

Fonte: O Autor (2025).

Os parâmetros B e Γ representam, respectivamente, a inclinação da reta e a suavidade da curva exponencial. Para problemas de classificação multiclases, como o abordado neste trabalho, utiliza-se a função de ativação *softmax* na camada de saída. Essa função converte o vetor de valores reais gerados pela rede em uma distribuição de probabilidade. Assim, a classe correspondente ao maior valor de probabilidade será atribuída ao padrão de entrada. Essa função é definida por:

$$\mathbf{y} = \mu(\mathbf{z}) = \frac{e^{z_k}}{\sum_{l=1}^m e^{z_l}}, \text{ para } k = 1, \dots, m \text{ e } \mathbf{z} = (z_1, \dots, z_m) \in \mathbb{R} \quad (4)$$

Em que m representa o número de nós (neurônios) na camada de saída.

O primeiro trabalho que trata sobre o aprendizado do neurônio biológico se credita a Hebb (1949), ao afirmar que esse aprendizado está relacionado com os estímulos (uso ou desuso) das sinapses, ou seja, a aprendizagem se dá pelo reforço das ligações sinápticas nos neurônios (Braga; Ludermir; Carvalho, 2007).

Mais tarde, o modelo Perceptron foi introduzido por Rosenblatt (1957). Ele é uma extensão do neurônio MCP, incorporando uma regra de aprendizado para ajustar as sinapses (pesos sinápticos) e detectar certos padrões. O Perceptron consiste em uma camada de entrada, uma camada intermediária e uma camada de saída com um único neurônio, sendo chamado de perceptron de única camada porque somente os pesos sinápticos entre a camada intermediária e

a camada de saída eram ajustáveis. Em outras palavras, os pesos entre a camada de entrada e a camada intermediária permaneciam fixos durante todo o processo de treinamento (Braga; Ludermir; Carvalho, 2007).

Considerando a função limiar e utilizando (3), pode ser observado que a condição mínima de disparo é:

$$\mathbf{w} \cdot \mathbf{x} = \theta \quad (5)$$

Ou

$$\mathbf{w} \cdot \mathbf{x} + w_0 = 0 \quad (6)$$

Em que o termo $w_0 = -\theta$ é conhecido como bias. Assim, considerando a unidade de *bias*, (3) pode ser reescrita como:

$$z = \mathbf{w} \cdot \mathbf{x} = \sum_{t=0}^T w_t x_t \quad (7)$$

Em que $\mathbf{x} = (1, x_1, x_2, \dots, x_T)$ e $\mathbf{w} = (w_0, w_1, w_2, \dots, w_T)$.

Sendo assim, para um vetor de entrada \mathbf{x} associado a uma saída desejada (rotulada) d , o erro do modelo pode ser determinado por:

$$e = d - y \quad (8)$$

E os pesos podem ser atualizados de acordo com:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \Delta \mathbf{w} = \mathbf{w}(n) + \eta e \mathbf{x} \quad (9)$$

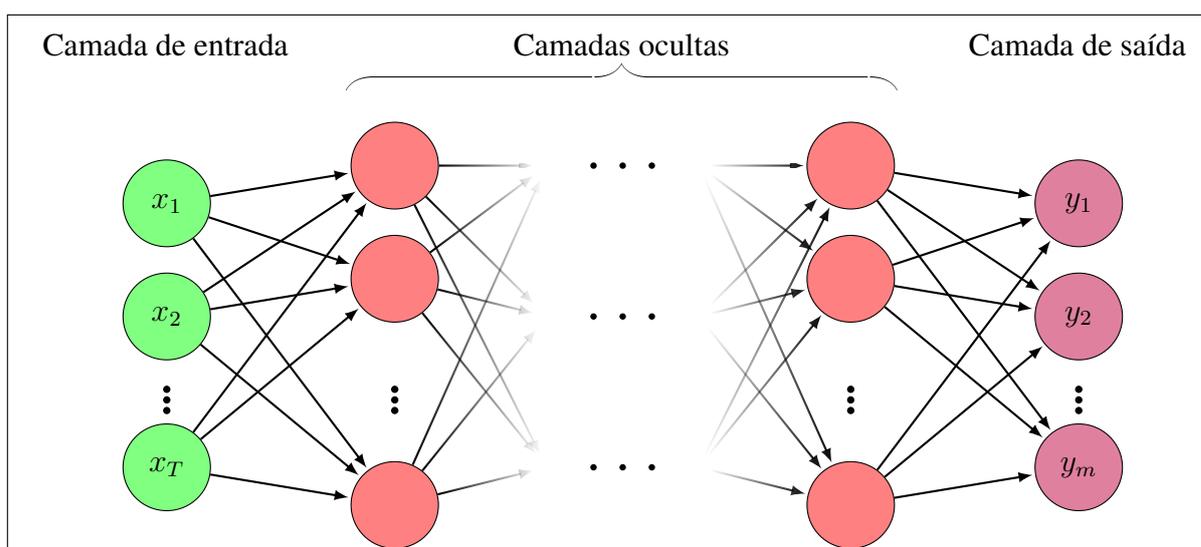
Em que $\mathbf{w}(n+1)$ representa o peso atualizado e η é a taxa de aprendizagem do algoritmo, que indica o quão rápido a atualização dos pesos será realizada, e possui um valor positivo menor que a unidade.

No entanto, esse modelo de única camada só trata de problemas linearmente separáveis. Nesse caso, converge em um número finito de passos. Mas, para os casos não linearmente separáveis, ele entra em *loop* indefinidamente sem convergir para a solução ótima.

Para resolver problemas não lineares, aplica-se o modelo MLP, que é uma rede com mais de uma camada com diversos neurônios que estão totalmente conectados entre si. Assim, são acrescentadas mais camadas intermediárias (ocultas) ao modelo e cada neurônio de uma camada anterior é conectado a todos os neurônios da camada posterior. Pode ser observado na

Figura 14 uma representação do MLP. Segundo Braga, Ludermir e Carvalho (2007), as camadas intermediárias são detectoras de padrões e com um número suficientemente grande dessas camadas é possível encontrar representação para qualquer conjunto de padrões de entrada. Foi matematicamente provado que um MLP com apenas uma camada escondida pode ser utilizado para aproximar qualquer função contínua (Cybenko, 1989) e com duas camadas qualquer função matemática (Cybenko, 1988). Segundo Heaton (2008), ainda não existem razões teóricas para se utilizar mais do que duas camadas intermediárias, porém o treinamento da rede pode se tornar mais fácil caso isso ocorra (Braga; Ludermir; Carvalho, 2007). Essas redes são do tipo *feedforward*, ou seja, ela processa os dados em um fluxo unidirecional (sem conexões de retroalimentação) com o objetivo de aproximar uma função (Goodfellow; Bengio; Courville, 2016).

Figura 14 – Representação do MLP.



Fonte: O Autor (2025).

3.1.1 Treinamento do MLP

O treinamento de um MLP é baseado na regra delta generalizada que utiliza a técnica do gradiente descendente (GD). Como deve-se separar dados que não podem ser separados por uma reta (ou um hiperplano no caso multidimensional), a função de ativação utilizada durante a aprendizagem deve ser não linear e diferenciável para que o gradiente possa ser calculado (Haykin, 2009).

O algoritmo de treinamento é conhecido como retropropagação do erro (*backpropagation*), que consiste de duas etapas:

- a) Etapa *forward*: O vetor de entrada x é aplicado na camada de entrada, seu efeito se

propaga para frente na rede e uma resposta y é obtida na camada de saída da rede. Durante essa etapa, os pesos sinápticos entre os neurônios de cada camada não são modificados;

- b) Etapa *backward*: Determina-se o erro dessa saída y em relação à saída desejada d e o propaga para trás na rede, atualizando os pesos.

Nesse algoritmo, o erro a ser minimizado (função de custo) é o erro quadrático entre a diferença do vetor da saída desejada d e o vetor da saída real da rede y :

$$E = \frac{1}{2} \sum_{l=1}^m e_l^2 = \frac{1}{2} \sum_{l=1}^m (d_l - y_l)^2 \quad (10)$$

Durante o treinamento com o *backpropagation*, uma amostra é apresentada à rede e propagada para frente até obter um resultado na camada de saída. Essa saída é comparada com a saída desejada e, se não estiver correta, é calculado o erro de acordo com (10).

Utiliza-se a técnica do gradiente descendente do erro em relação aos pesos para efetuar a atualização desses (também conhecida na literatura como regra delta generalizada). Considerando o peso w_{rq} entre os nós q e r nas camadas Q que antecede R , a variação desse deve ser proporcional ao negativo do gradiente, ou seja:

$$\Delta w_{rq} \propto -\frac{\partial E}{\partial w_{rq}} \quad (11)$$

O sinal negativo indica que o vetor de pesos sinápticos será deslocado no sentido que diminua o erro. Considerando a taxa de aprendizagem η , (11) pode ser reescrita como:

$$\Delta w_{rq} = -\eta \frac{\partial E}{\partial w_{rq}} \quad (12)$$

Através de manipulações algébricas e cálculos de derivadas parciais em (12), obtém-se a equação abaixo que determina a variação dos pesos entre os nós q e r :

$$\Delta w_{rq} = \eta \delta_r x_q \quad (13)$$

Considerando que R possui N_R neurônios, e caso ela seja uma camada intermediária, o valor de δ_r será determinado pela Equação (14) (em que N_P é o número de neurônios da camada P situada após a camada R) e caso seja uma camada de saída, será determinado por (15).

$$\delta_r = \mu'(z_r) \sum_{p=1}^{N_P} \delta_p w_{pr}, \quad \text{neurônio } r \text{ na camada intermediária} \quad (14)$$

$$\delta_r = (d_r - y_r)\mu'(z_r), \text{ neur\u00f4nio } r \text{ na camada de sa\u00edda} \quad (15)$$

Em que z_r representa o somat\u00f3rio do produto entre as entradas x e os pesos w , entre as camadas Q e R , ou seja:

$$z = \sum_{q=1}^{N_R} w_{rq}x_q \quad (16)$$

E, por fim, os pesos entre os neur\u00f4nios q e r s\u00e3o atualizados de acordo com:

$$w_{rq}(n+1) = w_{rq}(n) + \Delta w_{rq} \quad (17)$$

Segundo Braga, Ludermir e Carvalho (2007), o algoritmo do gradiente descendente tem algumas defici\u00eancias em problemas mais complexos, como o de converg\u00eancia para m\u00ednimos locais. Para tentar evitar esse problema, o termo de momento, ψ , \u00e9 adicionado \u00e0 Equa\u00e7\u00e3o (17), e ela assume a seguinte forma:

$$w_{rq}(n+1) = w_{rq}(n) + \Delta w_{rq} + \psi = w_{rq}(n) + \eta\delta_r x_q + \psi_\alpha[w_{rq}(n) - w_{rq}(n-1)] \quad (18)$$

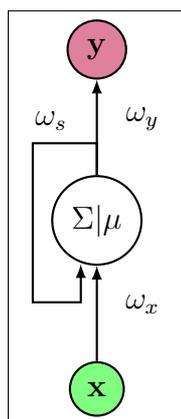
O gradiente descendente \u00e9 uma t\u00e9cnica fundamental de otimiza\u00e7\u00e3o que recebe diferentes denomina\u00e7\u00f5es dependendo do momento em que os pesos s\u00e3o atualizados. Quando as atualiza\u00e7\u00f5es ocorrem ap\u00f3s todo o conjunto de treinamento ter sido processado pelo modelo (uma \u00e9poca), \u00e9 chamado de Gradiente Descendente por Lotes (*Batch Gradient Descent* - BGD). Se os pesos s\u00e3o atualizados ap\u00f3s cada amostra do conjunto de treinamento, \u00e9 conhecido como Gradiente Descendente Estoc\u00e1stico (*Stochastic Gradient Descent* - SGD). Por fim, quando os pesos s\u00e3o atualizados ap\u00f3s um n\u00famero fixo de amostras (mini-lotes ou *batch*), \u00e9 denominado Gradiente Descendente por Mini-lotes (*Mini-Batch Gradient Descent* - MBGD).

O BGD \u00e9 computacionalmente mais eficiente, pois realiza poucas atualiza\u00e7\u00f5es de peso, resultando em um gradiente mais est\u00e1vel. No entanto, essa estabilidade pode levar a uma converg\u00eancia prematura para pesos menos \u00f3timos. Por outro lado, o SGD \u00e9 mais oneroso computacionalmente devido \u00e0s suas frequentes atualiza\u00e7\u00f5es de peso, o que pode gerar erros com maior vari\u00e2ncia (mais ruidosos). Essa instabilidade pode auxiliar na converg\u00eancia prematura em um m\u00ednimo local. O MBGD busca um equil\u00edbrio entre o BGD e o SGD, sendo amplamente utilizado na pr\u00e1tica, atualmente. Um estudo realizado por Bengio (2012) sugere que 32 pode ser um valor ideal como padr\u00e3o. Os tamanhos de mini-lotes geralmente s\u00e3o pot\u00eancias de dois que atendem aos requisitos de mem\u00f3ria do computador, como 32, 64, 128 ou 256.

3.2 LONG SHORT-TERM MEMORY - LSTM

Ao contrário do MLP, a rede LSTM é bidirecional (recorrente), o que envolve conexões de retroalimentação e a torna adequada para lidar com dados sequenciais, como séries temporais ou processamento de linguagem natural. Ela é uma variação de Rede Neural Recorrente (*Recurrent Neural Network* - RNN). A Figura 15 contém a estrutura básica de uma RNN (desconsiderando os termos de bias com fins de simplificação). Nela, y representa o vetor de saída que contém as previsões, enquanto ω_x , ω_y e ω_s são, respectivamente, os pesos associados à entrada, à saída e ao estado oculto da rede. O neurônio de uma RNN é parecido com o de um MLP, mas com a distinção de que seu estado oculto também serve como entrada para ele mesmo, introduzindo o conceito de memória na rede, permitindo que ela retenha informações de momentos anteriores. Assim, as RNN incorporam elementos de um MLP enquanto exploram sua capacidade de resolver problemas não lineares (Haykin, 2009).

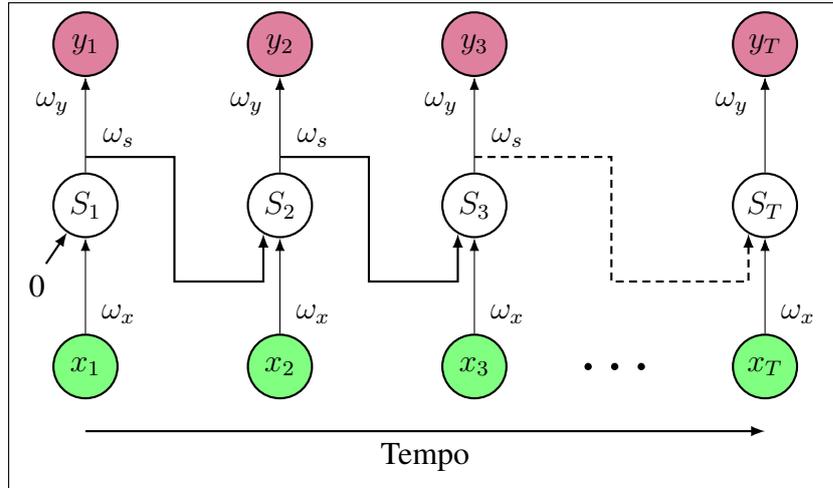
Figura 15 – Estrutura básica de uma RNN.



Fonte: O Autor (2025).

Uma maneira alternativa de visualizar este único neurônio é de acordo com a Figura 16, em que S_t representa o estado da rede no instante t . Em cada instante de tempo $t \leq T$, o neurônio recorrente recebe a entrada x_t e o estado do instante anterior S_{t-1} . Como não há informação antes do instante inicial, normalmente considera-se que a entrada recorrente é zero neste primeiro momento (Geron, 2022).

Figura 16 – RNN básica desdobrada ao longo do tempo.



Fonte: O Autor (2025).

Para uma RNN, o estado e a saída em um determinado instante t são determinados pelas Equações (19) e (20), em que b_s e b_y representam, respectivamente, os *bias*.

$$S_t = \mu(z) = \mu(w_s S_{t-1} + w_x x_t + b_s) \quad (19)$$

$$y_t = w_y S_t + b_y \quad (20)$$

Observa-se na Figura 16 que os pesos são compartilhados em cada instante de tempo, reduzindo o número de parâmetros que precisam ser treinados e consequentemente fazendo com que uma RNN possa ser utilizada para realizar previsões em sequências/séries de comprimento variável.

Porém, uma RNN convencional possui deficiência na tarefa de prever dependências a longo prazo, devido ao fato do problema da explosão ou desaparecimento do gradiente (Goodfellow; Bengio; Courville, 2016). Por exemplo, considerando neurônios com funções de ativação ReLU, se o peso w_s dos estados ocultos for maior que 1, à medida que a RNN é desdobrada ao longo do tempo, a saída no estado seguinte será amplificada por esse valor de peso, resultando em um fator de w_s^T no último estado. Isso pode elevar consideravelmente o valor da saída, o que cria um problema de convergência para o algoritmo *backpropagation*, que dará passos muito grandes durante o treinamento e terá dificuldade em encontrar o ponto ótimo dos pesos (explosão do gradiente). O problema de desaparecimento do gradiente ocorre de maneira análoga quando o peso w_s é menor do que 1. Nesse caso, a saída da RNN produz valores muito pequenos, resultando em valores de gradientes desprezíveis. Consequentemente, os passos dados

pelo algoritmo para ajustar os pesos serão extremamente reduzidos, dificultando a convergência para o ponto ótimo. As funções sigmoideal e tangente hiperbólica também podem enfrentar o problema do desaparecimento do gradiente em redes neurais recorrentes, assim como em redes de *Deep Learning* (DL) que naturalmente possuem muitas camadas ocultas (Raschka; Mirjalili, 2019). Esse problema ocorre quando as entradas dessas funções geram saídas próximas de 0 ou 1, resultando em derivadas quase nulas e, conseqüentemente, no desaparecimento do gradiente.

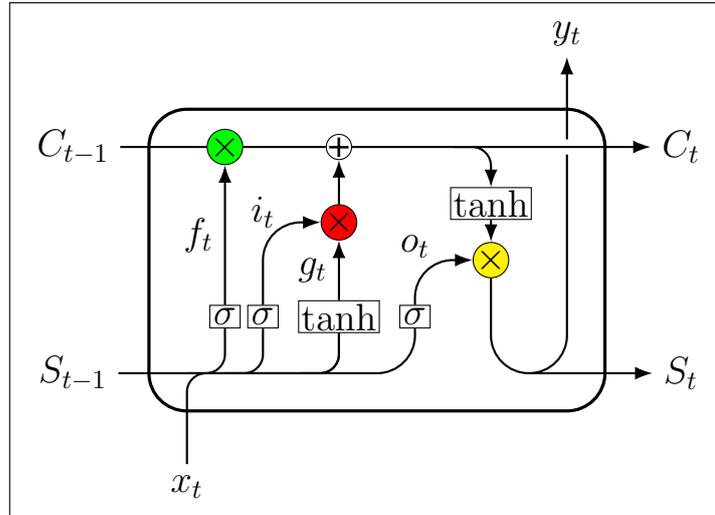
A LSTM foi projetada com o intuito de resolver esse problema que ocorre em RNNs convencionais quando são desdobradas em longos períodos de tempos (Hochreiter; Schmidhuber, 1997). A ideia por trás da LSTM é que, em vez de usar a mesma conexão de retroalimentação para eventos ocorridos há muito tempo e eventos mais recentes (como em uma RNN convencional com peso sináptico igual), ela emprega conexões distintas para lidar com eventos de longo prazo e aqueles mais recentes. Isso é alcançado mediante a inclusão de portas que regulam o fluxo de informações e a atualização dos gradientes.

Na Figura 17, é apresentada a estrutura fundamental (célula) da LSTM, com suas respectivas portas. Os valores de entrada e saída, além dos estados ocultos e estados da célula, são representados como x_t , y_t , S_t e C_t , respectivamente. Observa-se que na LSTM, o valor da saída é sempre igual ao valor do estado oculto, ou seja:

$$y_t = S_t \quad (21)$$

A LSTM é dividida em três portas, esquecimento (verde), entrada (vermelha) e saída (amarela). A função de ativação sigmoideal que alimenta a porta de esquecimento através de f_t decide quais informações serão mantidas (um valor de 0 descarta todas as informações, enquanto 1 mantém todas). Outra função sigmoideal em conjunto com a função tangente hiperbólica fornecem os valores i_t e g_t para a porta de entrada que seleciona novas informações que serão adicionadas ao estado da célula C_{t-1} para produzir um novo estado atualizado C_t . Finalmente, esse estado atualizado C_t é aplicado a outra função tangente hiperbólica e multiplicado por o_t com o intuito de realizar uma filtragem na porta de saída para determinar quais informações serão emitidas como saída y_t no passo de tempo atual, e também servirá como estado oculto S_t para o próximo passo de tempo.

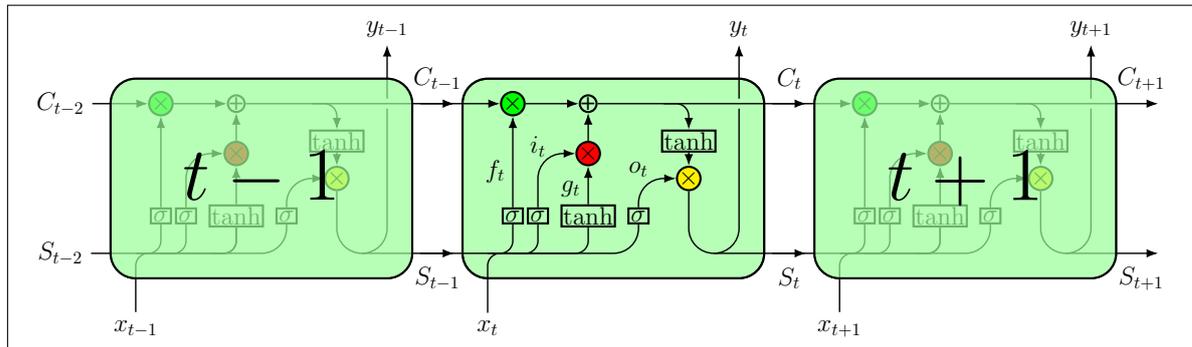
Figura 17 – Célula da LSTM.



Fonte: Adaptado de Souza *et al.* (2024).

A Figura 18 é a rede LSTM desdobrada ao longo do tempo de forma similar como foi feito com a RNN mostrada anteriormente.

Figura 18 – LSTM desdobrada ao longo do tempo.



Fonte: O Autor (2025).

Observando a Figura 17, os valores de f_t , i_t , g_t e o_t podem ser calculados como segue (Graves; Mohamed; Hinton, 2013).

$$f_t = \sigma(w_{sf}S_{t-1} + w_{xf}x_t + b_f) \quad (22)$$

$$i_t = \sigma(w_{si}S_{t-1} + w_{xi}x_t + b_i) \quad (23)$$

$$g_t = \tanh(w_{sg}S_{t-1} + w_{xg}x_t + b_g) \quad (24)$$

$$o_t = \sigma(w_{so}S_{t-1} + w_{xo}x_t + b_o) \quad (25)$$

Em que:

- a) σ : Função de ativação sigmoidal;
- b) $w_{sf}, w_{si}, w_{sg}, w_{so}$: Pesos do estado oculto aplicados às portas de esquecimento, entrada e saída, respectivamente;
- c) $w_{xf}, w_{xi}, w_{xg}, w_{xo}$: Pesos da entrada aplicados às portas de esquecimento, entrada e saída, respectivamente;
- d) b_f, b_i, b_g, b_o : *Bias* aplicados às portas de esquecimento, entrada e saída, respectivamente.

Por fim, o estado da célula e sua saída podem ser calculados de acordo com as Equações (26) e (27), respectivamente.

$$C_t = f_t C_{t-1} + i_t g_t \quad (26)$$

$$y_t = S_t = o_t \tanh(C_t) \quad (27)$$

O valor de C_t é considerado como a memória de longo prazo, ou seja, ela armazena informações ao longo de toda a sequência de entrada sem ser alterado por pesos sinápticos. Da Figura 17, observa-se que a medida que C_{t-1} flui pela célula da esquerda para a direita, algumas informações são removidas pela porta de esquecimento e outras são adicionadas pela porta de entrada, e assim C_t é obtido sem mais nenhuma modificação. Por outro lado, o valor de S_t representa a memória de curto prazo que é alterada por pesos sinápticos, refletindo o estado atual da rede. O valor de C_t é copiado e fornecido a uma função de ativação \tanh que é posteriormente filtrado pela porta de saída para produzir esse estado atual S_t (Geron, 2022).

4 METODOLOGIA

Neste capítulo, serão abordados com maior detalhamento os seguintes aspectos: a estrutura da base de dados, o procedimento de pré-processamento para remoção de dados inconsistentes, os atributos extraídos das séries temporais de consumo de energia elétrica, os modelos desenvolvidos e a metodologia selecionada para avaliar o desempenho deles.

Todos os códigos foram desenvolvidos no software *Spyder*, através da linguagem de programação *Python* em um *desktop* com sistema operacional *Windows 11*, processador *Intel Core i7-12700* da 12ª geração, 16 GB de memória RAM e uma placa de vídeo *Nvidia GeForce RTX 3060* com 12 GB de memória dedicada. Os modelos foram criados utilizando o *Keras*, uma biblioteca de alto nível, escrita em *Python* para construir e treinar redes neurais. Foi utilizada uma base de dados aberta de consumidores da Irlanda, criada pela *Commission for Energy Regulation (CER)* e administrada pelo *Irish Social Science Data Archive (ISSDA, 2010)*.

4.1 BASE DE DADOS

Essa base é fornecida pelo ISSDA após solicitação por qualquer usuário apenas para fins educacionais ou de pesquisa. Ela inclui medições diárias de consumo de energia elétrica em kWh, com uma taxa de amostragem semi-horária (48 medições por dia), de mais de 6.400 consumidores. Esse total abrange consumidores residenciais, comerciais, industriais e serviços públicos, ao longo de 535 dias, entre os anos de 2009 e 2010.

No presente estudo, foram selecionados aleatoriamente 650 consumidores residenciais para formar a base de dados destinada ao treinamento e teste dos modelos desenvolvidos. Foram realizadas etapas de pré-processamento para garantir a consistência dos dados utilizados, já que os modelos de IA são altamente sensíveis às informações fornecidas.

Existem diversos tipos de erros que podem ocorrer em bases de dados, como a ausência ou a duplicação dos dados. Para lidar com dados ausentes, algumas abordagens comuns incluem a imputação (preenchimento com base em outras observações da mesma base), a interpolação (utilização de dados vizinhos para estimar o valor faltante) ou a exclusão dos períodos em que o problema ocorre (Nielsen, 2021). Para tratar dados duplicados, geralmente remove-se apenas o valor redundante.

Três tipos de erros foram encontrados e tratados de forma similar ao que foi realizado em (Souza *et al.*, 2024):

- a) Se uma ou mais medições semi-horárias estiverem ausentes em um determinado dia, esse

dia inteiro é excluído;

- b) Se forem encontrados dados incompatíveis, como ter mais de 48 medições em um único dia, o dia inteiro é excluído;
- c) Alguns consumidores apresentaram medições iguais a zero durante várias horas em determinados dias, o que pode indicar uma falha no medidor nesses momentos. Portanto, se um consumidor tiver mais de 1/4 do dia (12 medições) com valores zerados, o dia inteiro é descartado.

Deve-ser ressaltar que usualmente não se removem amostras da base, a fim de evitar perda de informação. Porém, como há uma quantidade considerável de dados na base em questão, a remoção de algumas amostras não afeta o treinamento dos modelos.

Considerando os instantes de tempo $t \in \{1, 2, \dots, T\}$, cada série temporal de consumo de energia é um vetor \mathbf{x} , ou seja:

$$\mathbf{x} = [x_1, x_2, \dots, x_T] \quad (28)$$

Como mencionado anteriormente, foram selecionados $C = 650$ consumidores residenciais. Cada consumidor teve seus dados coletados ao longo de $D = 535$ dias, gerando $N = CD = 347.750$ séries temporais de consumidores honestos. Sendo assim, a matriz \mathbf{X} com dimensões $N \times T$, contendo as séries de todos os consumidores pode ser definida como:

$$\mathbf{X} \in \mathbb{R}^{N \times T} \quad (29)$$

Essa matriz assume o seguinte formato:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_{N-1} \\ \mathbf{x}_N \end{bmatrix} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,T} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,T} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N-1,1} & x_{N-1,2} & \cdots & x_{N-1,T} \\ x_{N,1} & x_{N,2} & \cdots & x_{N,T} \end{bmatrix} \quad (30)$$

4.2 MODELAGEM DAS PNT

Os dados fornecidos pela ISSDA são considerados de consumidores honestos, uma vez que eles concordaram em participar do processo de instalação do medidor inteligente em suas residências. Com base nisso, são necessárias equações para modelar o comportamento de consumo

de usuários fraudulentos ou de medidores defeituosos. Neste trabalho, são utilizadas as mesmas 11 equações trabalhadas por Souza *et al.* (2024), que são equações modificadas do trabalho de Chuwa e Wang (2021), visto que elas cobrem um grande número de possibilidades de PNT. Essas modificações são realizadas com o intuito de gerar casos mais práticos de furto de energia. Com o intuito de balancear a base de dados, para cada amostra de (30), foi gerada uma amostra de cada modelo de PNT. Considere \mathbf{m}_f , em que f varia de 1 a 11, representando as onze possibilidades de amostras maliciosas (tipos de ataques/fraudes cometidos contra os medidores inteligentes) para determinado consumidor com padrão \mathbf{x} .

- a) $f = 1$: A série temporal é multiplicada por um fator aleatório α entre 0,1 e 0,9 que reduz o consumo de energia elétrica:

$$\mathbf{m}_1 = \alpha \mathbf{x} \quad (31)$$

- b) $f = 2$: Cada medidor inteligente registra consumo zero enquanto o ataque ocorre (*bypass*/contorno total do medidor). As variáveis t_s e t_e representam, respectivamente, os horários de início e fim do ataque ao medidor. Neste trabalho, a menos que indicado o contrário, todos os ataques que ocorrem durante um período começam aleatoriamente entre 08:00:00 e 16:00:00 e duram 4 horas. Por exemplo, se o ataque ao medidor começou às 08:00:00, ele perdura até às 12:00:00:

$$\mathbf{m}_2 = \beta \mathbf{x}, \quad \beta = \begin{cases} 0, & \text{se } t_s < t < t_e \\ 1, & \text{caso contrário} \end{cases} \quad (32)$$

- c) $f = 3$: Este tipo de ataque é semelhante à fraude $f = 1$, mas, em vez de multiplicar toda a série temporal por um único valor aleatório, as medições na série temporal em cada instante de tempo t são multiplicadas por um valor aleatório diferente, γ_t , entre 0,1 e 0,9:

$$\mathbf{m}_3 = \gamma_t \mathbf{x} \quad (33)$$

- d) $f = 4$: É um tipo de fraude semelhante ao ataque $f = 2$, mas em vez de contornar completamente o medidor, ocorre um contorno parcial:

$$\mathbf{m}_4 = \kappa \mathbf{x}, \quad \kappa = \begin{cases} \alpha, & t_s < t < t_e \\ 1, & \text{caso contrário} \end{cases} \quad (34)$$

- e) $f = 5$: O consumo real é substituído pelo produto entre o consumo médio, $\bar{\mathbf{x}}$, e diferentes valores aleatórios:

$$\mathbf{m}_5 = \gamma_t \bar{\mathbf{x}} \quad (35)$$

f) $f = 6$: Um ponto de corte, co é selecionado. A medição da série temporal em cada instante de tempo t é substituída pelo ponto de corte se for maior do que ele. O ponto escolhido nesse trabalho para todos os tipos de fraudes que o utilizam é um valor aleatório entre 20% e 30% acima do consumo médio de energia:

$$\mathbf{m}_6 = \begin{cases} x_t, & x_t \leq co \\ co, & x_t > co \end{cases} \quad (36)$$

Ou seja, $co = (1 + \frac{r}{10}) \bar{x}$ e r é um valor aleatório no intervalo [2, 3].

g) $f = 7$: Um ponto de corte, co , é selecionado. Subtrai-se cada medição da série temporal por co . Se o resultado é menor do que zero, é retornado um consumo igual a zero. Ou seja, determina-se sempre o valor máximo entre a diferença da série \mathbf{x} pelo valor de corte co e zero:

$$\mathbf{m}_7 = \max(\mathbf{x} - co, 0) \quad (37)$$

h) $f = 8$: Diferente de outros tipos de ataques, este não causa uma queda abrupta no consumo de energia. Em vez disso, a redução ocorre de maneira linear ao longo do tempo até alcançar a intensidade máxima do ataque i_{max} . Esse declínio gradual é determinado pela taxa de variação da intensidade do ataque, que corresponde à inclinação s :

$$\mathbf{m}_8 = (1 - i)\mathbf{x}, i = \begin{cases} i_{max}, & t \geq t_e \\ s(t - t_s), & t_s \leq t < t_e \\ 0, & t < t_s \end{cases} \quad (38)$$

Em que $s = \frac{i_{max} - i_{min}}{t_e - t_s}$, $i_{min} = 0$ e $i_{max} = 0,9$.

i) $f = 9$: Cada série temporal, \mathbf{x} , é substituída pelo seu valor médio:

$$\mathbf{m}_9 = \bar{\mathbf{x}} \quad (39)$$

j) $f = 10$: O padrão de consumo de energia se revertido ao longo do tempo. Esse tipo de ataque acontece em contextos em que o preço da energia varia ao longo do dia. Por exemplo, um usuário que consome mais e paga uma tarifa maior durante o período noturno, ao reverter seu padrão de consumo, poderá ver sua conta de energia diminuída:

$$\mathbf{m}_{10} = \mathbf{x}_{T-t+1} \quad (40)$$

Sendo $\mathbf{x}_{T-t+1} = [x_T, x_{T-1}, \dots, x_1]$.

k) $f = 11$: Um outro tipo de ataque busca tirar proveito das variações nas tarifas de energia ao longo do dia. O consumo é diminuído apenas em horários específicos, como durante os picos de demanda, quando as tarifas estão elevadas, e é redistribuído para períodos em que as tarifas são mais baixas. Assim, o total de energia consumida pelo cliente se mantém constante ao longo do dia:

$$\mathbf{m}_{11} = \begin{cases} f_r \mathbf{x}, & t_s \leq t < t_e \\ \mathbf{x} + \frac{\delta_E}{(T - N_a)}, & \text{caso contrário} \end{cases} \quad (41)$$

Aqui, $f_r = 0.3$ é o fator de redução. Para este ataque específico, que visa simular consumidores residenciais que cometem fraudes durante a noite, o horário de início, t_s , é determinado aleatoriamente entre 18:00:00 e 20:00:00, e a finalização ocorre em qualquer momento subsequente. N_a é o número de instantes de tempo em que o ataque ocorre e δ_E indica a quantidade total de energia que foi reduzida no processo.

4.3 REDUÇÃO DE DIMENSIONALIDADE

Bases de dados com alta dimensionalidade, ou seja, aquelas em que o número de atributos é muito maior que o número de amostras (Narisetty, 2020), são geralmente problemáticos para os algoritmos de IA devido ao alto custo computacional e uso de memória (Janecek *et al.*, 2008).

Existem duas abordagens para a redução de dimensionalidade de uma base, seleção de atributos e extração de atributos (Khalid; Khalil; Nasreen, 2014). Na seleção de atributos, apenas aqueles que possuem informações mais importantes/relevantes são selecionados, baseados em determinado critério para servir como entrada do modelo.

Um método simples e computacionalmente eficiente é a seleção através de algum filtro, como o filtro de *low variance*. Como o nome sugere, é aplicado um filtro que irá remover os atributos que possuem uma variância abaixo de um valor estipulado (*Threshold*). A ideia por trás dessa técnica é que um atributo que possui baixo valor de variância (ou seja, dados que não estão muito espalhados em relação à média), não agrega no processo de aprendizagem do algoritmo. O caso extremo seria um atributo que possui o mesmo valor em todas as mostras, gerando uma variância igual a zero.

A extração de atributos é uma abordagem mais abrangente que visa transformar todo o espaço das variáveis originais em um subespaço de dimensão reduzida, preservando a maior parte das informações relevantes. Esse processo pode ser realizado de forma direta, como calcular a média de uma série temporal, ou por meio de técnicas específicas, como PCA, t -

Distributed Stochastic Neighbor Embedding (t-SNE) ou UMAP. Essas técnicas projetam os atributos do espaço original para um subespaço de dimensões reduzidas, sendo que o PCA faz isso de maneira linear, enquanto t-SNE e UMAP utilizam projeções não lineares.

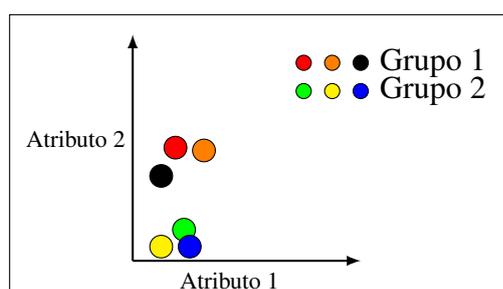
Embora a base de dados utilizada neste trabalho não apresente alta dimensionalidade (já que o número de amostras é significativamente maior que o número de instantes em que cada amostra é medida), é viável aplicar qualquer método de redução de dimensionalidade para potencializar o desempenho preditivo dos modelos. Além de usar os atributos resultantes da redução como entradas, técnicas como t-SNE e UMAP são frequentemente empregadas para visualização, reduzindo a dimensionalidade de T para 2 ou 3 dimensões, e auxiliando na identificação de padrões nos dados em diversos campos científicos (Maaten; Hinton, 2008; Chuwa; Wang, 2021; Zarzà; Curtò; Calafate, 2023), inclusive em aplicações de séries temporais (Ali *et al.*, 2019).

4.3.1 *Uniform Manifold Approximation and Projection - UMAP*

O UMAP foi escolhido como o método para a extração de atributos neste trabalho, pois tem se destacado em diversos testes de desempenho em várias bases de dados e é considerado o estado da arte no que tange a redução de dimensionalidade (Zarzà; Curtò; Calafate, 2023). Esses testes compararam o desempenho do UMAP com outros métodos amplamente utilizados na literatura, tanto em termos do tempo necessário para realizar a redução da dimensionalidade quanto nos resultados obtidos pelos classificadores ao utilizar os atributos extraídos (McInnes; Healy; Melville, 2020).

Considerando uma base de dados com dois atributos e dois agrupamentos (*clusters*), conforme a Figura 19, é possível reduzir a dimensão desses dados do espaço bidimensional para um espaço unidimensional. Duas formas de realizar essa redução podem ser observadas na Figura 20.

Figura 19 – Dados em dois agrupamentos distintos.



Fonte: O Autor (2025).

Figura 20 – Redução de dimensionalidade.



Fonte: O Autor (2025).

Ao realizar uma projeção linear das amostras no eixo das abscissas, ocorre uma mistura dos pontos, o que pode prejudicar o aprendizado de determinados algoritmos, uma vez que os valores dos atributos se tornam semelhantes entre todos os grupos considerados (Figura 20a). O UMAP (Figura 20b), por sua vez, projeta os dados no espaço de dimensão inferior e, por meio de um processo iterativo, move essas amostras preservando as relações que tinham no espaço de maior dimensão. Assim, o algoritmo garante que as amostras do grupo 1 permaneçam próximas umas das outras (preservando a relação local dos dados), assim como as amostras do grupo 2, que também se mantêm unidas, enquanto os dois grupos ficam relativamente distantes entre si (preservando a relação global dos dados). Durante esse processo, são calculadas pontuações de similaridade para facilitar o agrupamento no espaço reduzido.

Deseja-se reduzir a dimensão dos dados para uma dimensão $\tau < T$. Esse é um processo de associação conhecido como *embedding*, em que serão criados novos vetores a partir das amostras dos dados originais. Para facilitar a visualização dos grupos, normalmente utiliza-se $\tau = 2$ ou $\tau = 3$ (Damrich; Hamprecht, 2021). No entanto, é possível optar por um valor maior para tentar extrair mais atributos e, assim, aprimorar o desempenho de um modelo preditivo.

O UMAP calcula pontuações de similaridade (p) entre as amostras/pontos da base dados na dimensão original (superior). Como o próprio nome indica, essa pontuação quantifica o quão semelhantes são dois pontos. Para realizar esse cálculo, inicialmente determinam-se os valores das conectividades locais ρ_i , para cada amostra da base de dados, ou seja, para $i = 1, 2, \dots, N$. A conectividade local representa a distância entre o ponto \mathbf{x}_i e o seu k -vizinho mais próximo, ou seja:

$$\rho_i = \min\{d(\mathbf{x}_i, \mathbf{x}_j) | 1 \leq j \leq k\} \quad (42)$$

Com base em uma distribuição exponencial, a pontuação de similaridade de um ponto \mathbf{x}_i em relação a um ponto \mathbf{x}_j é determinada da seguinte forma:

$$p_{ij} = e^{-[d(\mathbf{x}_i, \mathbf{x}_j) - \rho_i] / \sigma_i} \quad (43)$$

Em que σ_i é um fator de suavização (escalonamento) da curva exponencial. Essa similaridade deve ser determinada para todos os vizinhos mais próximos, ou seja, para $1 \leq j \leq k$. No caso em que o vizinho mais próximo seja o próprio ponto \mathbf{x}_j , haveria uma distância igual a zero, resultando em uma pontuação unitária, ou seja, o maior valor possível de similaridade.

O UMAP ajusta a noção local de similaridade para cada ponto \mathbf{x}_i ao selecionar o fator de suavização σ_i . Esse fator é escolhido de forma que a similaridade total (a soma das similaridades em relação a cada um dos k vizinhos mais próximos) seja normalizada. Em outras palavras, conforme Gularte (2023), por meio de uma busca binária, ele deve encontrar σ_i de modo que:

$$\sum_{j=1}^k e^{-[d(\mathbf{x}_i, \mathbf{x}_j) - \rho_i]/\sigma_i} = \log_2(k) \quad (44)$$

No entanto, para assegurar a igualdade em (44), em que k é um valor fixo, diferentes valores de σ_i são gerados para cada ponto x_i . Isso resulta em distribuições exponenciais distintas e, conseqüentemente, em pontuações de similaridade assimétricas (por exemplo, a p_{ij} pode ser diferente da p_{ji}). Para converter todas as pontuações de modo que fiquem simétricas, a seguinte operação é realizada:

$$p_{ij} = p_{ji} = p_{ij} + p_{ji} - p_{ij}p_{ji} \quad (45)$$

A Equação (45) define a similaridade na dimensão original (superior) dos dados.

A seguir, o UMAP projeta os dados para a dimensão inferior através da técnica *Spectral Embedding* (McInnes; Healy; Melville, 2020) e utiliza uma aproximação da função de similaridade da dimensão superior (Equação (43)) para calcular as similaridades na dimensão inferior, q_{ij} :

$$q_{ij} = \frac{1}{1 + a d_{inf}(\mathbf{x}_i, \mathbf{x}_j)^{2b}} \approx \begin{cases} 1, & d_{inf}(\mathbf{x}_i, \mathbf{x}_j) \leq \delta \\ e^{-[d_{inf}(\mathbf{x}_i, \mathbf{x}_j) - \delta]}, & d_{inf}(\mathbf{x}_i, \mathbf{x}_j) > \delta \end{cases} \quad (46)$$

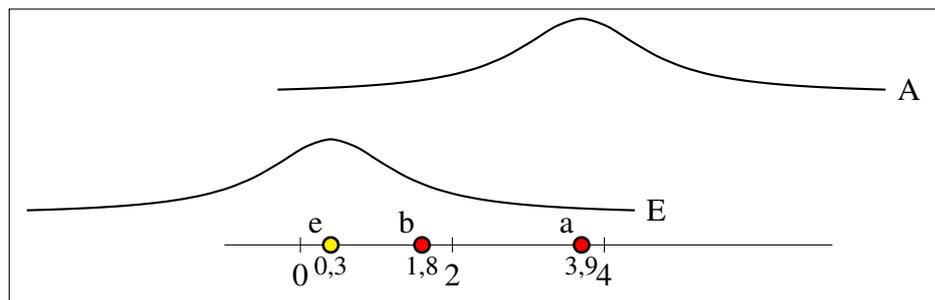
Através do método dos mínimos quadrados não linear, o UMAP determina os valores de a e b baseado no valor fornecido do hiperparâmetro δ , que é a distância mínima entre dois pontos na dimensão inferior desejada pelo usuário. O valor padrão é $\delta = 0,1$, resultando em $a = 1,577$ e $b = 0,895$; $d_{inf}(\mathbf{x}_i, \mathbf{x}_j)$ representa a distância entre os pontos \mathbf{x}_i e \mathbf{x}_j na dimensão inferior.

A Equação (46) é similar a uma distribuição de Cauchy, que é um caso especial da distribuição t . É baseado nessa curva de distribuição que o UMAP irá mover os pontos no espaço de dimensão inferior para que eles mantenham a relação local e global que tinham na dimensão original dos dados.

Para reproduzir os grupos iniciais da dimensão superior, o UMAP seleciona um par de pontos, x_i e x_j , na dimensão inferior e os ajusta para ficarem mais próximos, preservando a relação local. Essa seleção é feita de forma aleatória, levando em conta os valores de p_{ij} (valores mais altos indicam maior probabilidade de escolha). Após essa seleção, decide-se aleatoriamente se o ponto x_i será movido em direção ao ponto x_j ou vice-versa. Em seguida, escolhe-se um ponto do qual ponto escolhido deve se afastar (que será algum ponto que não estará entre os k -vizinhos mais próximos) para preservar a relação global. Por fim, minimiza-se a função de custo para encontrar a melhor posição, indicando como os pontos estarão agrupados na dimensão inferior.

Por exemplo, considere o caso da Figura 21 em que existem três pontos: a, b e e que já foram projetados em uma dimensão inferior. Os pontos a e b pertencem ao mesmo grupo (vermelho), enquanto o ponto e é de um grupo distinto (amarelo). Portanto, o UMAP deve ajustar os pontos para que b se aproxime de a, ao mesmo tempo em que se distancie de e.

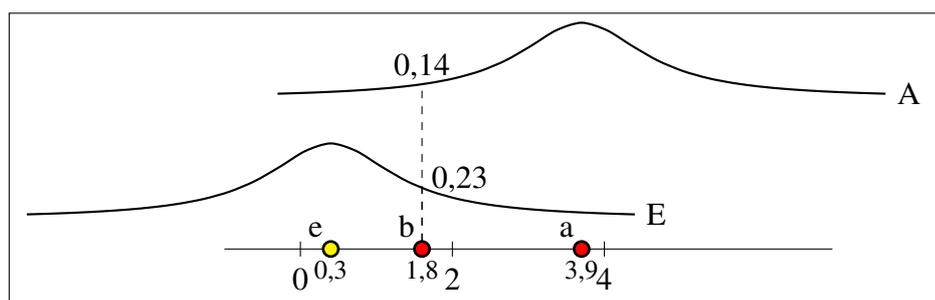
Figura 21 – Inicialização dos pontos na dimensão inferior.



Fonte: O Autor (2025).

Utilizando a Equação (46), obtém-se uma similaridade de 0,14 entre os pontos b e a, e uma similaridade de 0,23 entre b e e, como observado na Figura 22. À medida que o ponto b se aproxima de a, sua similaridade aumenta (a curva A apresenta um aumento), enquanto sua similaridade em relação ao ponto e diminui (a curva E apresenta uma redução).

Figura 22 – Similaridade na dimensão inferior.



Fonte: O Autor (2025).

Por fim, para realizar a avaliação da similiaridade, o UMAP minimiza a seguinte função de custo C :

$$C = - \sum_{i \neq j} [p_{ij} \ln(q_{ij}) + (1 - p_{ij}) \ln(1 - q_{ij})] \quad (47)$$

Ela representa a entropia cruzada entre as probabilidades (similaridades) da dimensão elevada p_{ij} e da dimensão inferior q_{ij} . Deseja-se minimizar a função de custo em relação às distâncias na dimensão inferior, para que a distribuição de pontos nela se matenha o mais próximo possível da distribuição do espaço original. Essa minimização é realizada através da técnica do SGD (Zarzà; Curtò; Calafate, 2023). A derivada do primeiro termo do somatório de (47) representa a atração que o ponto x_i exerce sobre o ponto x_j em razão da similaridade na dimensão original p_{ij} . Por outro lado, a derivada do segundo termo reflete a repulsão entre esses dois pontos, causada pela ausência de similaridade nessa mesma dimensão, representada por $(1 - p_{ij})$ (Damrich; Hamprecht, 2021).

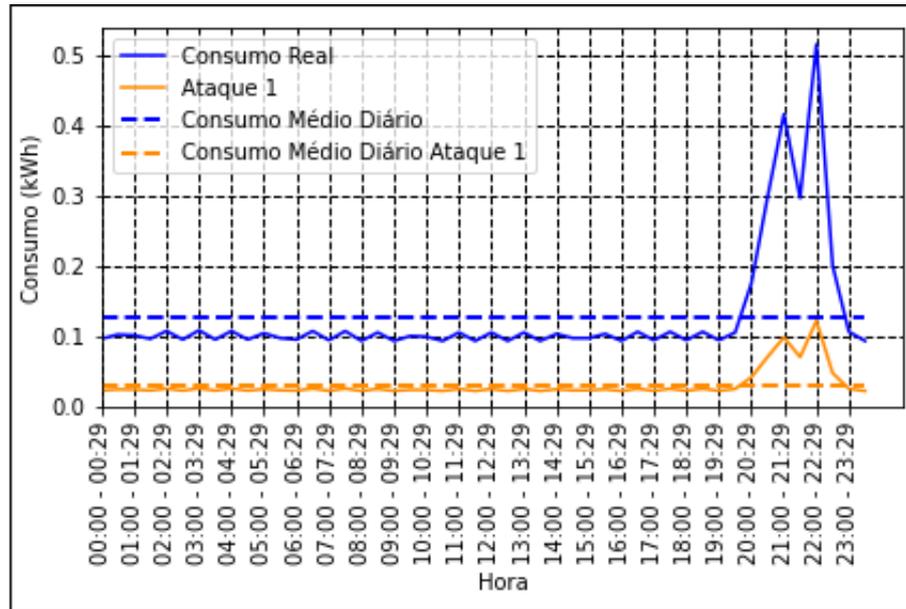
4.3.2 Extração de Atributos

Alguns tipos de fraudes são mais difíceis de identificar do que outros. Isso ocorre principalmente porque um consumidor que rouba uma quantidade pequena de energia pode ser erroneamente classificado como um consumidor de baixo consumo (Zanetti *et al.*, 2019; Chuwa; Wang, 2021).

A partir dessa informação, os ataques que apresentam maior dificuldade de detecção são os tipos 1, 3 e 4, quando a fraude é realizada com altos valores de α , γ_t e κ , respectivamente. Assim, atributos ou características adicionais podem ser utilizados para aprimorar o desempenho de detecção de um modelo, especialmente nesses casos.

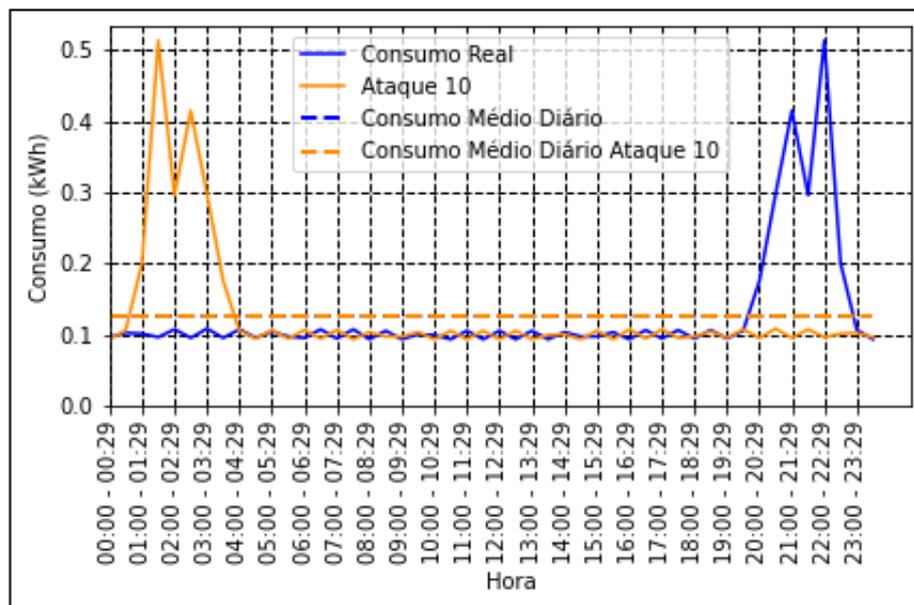
Foram extraídos atributos estatísticos (média e variância) e temporais (coordenadas do centroide), inicialmente utilizados por Souza *et al.* (2024). Além disso, neste trabalho, também foram empregados os atributos gerados pela técnica UMAP.

- a) Média: A maior parte das fraudes resulta em uma diminuição do consumo de energia, o que leva a concluir que o consumo energético de um usuário fraudulento tende a ser inferior ao de um usuário honesto. Dessa forma, o valor médio pode fornecer informações úteis para a detecção de fraudes. Por exemplo, conforme a Figura 23, o valor médio de um usuário que comete fraude do tipo $f = 1$ é inferior ao seu consumo real, o que pode ajudar a melhorar a detecção desse tipo de ataque.

Figura 23 – Consumo real e fraude do tipo $f = 1$.

Fonte: O Autor (2025).

No entanto, para um ataque do tipo $f = 10$, em que a fraude ocorre pela inversão do padrão de consumo de energia, o valor médio permanece inalterado, como pode ser observado na Figura 24. Assim, é necessário incluir características adicionais para aprimorar o processo de classificação.

Figura 24 – Consumo real e fraude do tipo $f = 10$.

Fonte: O Autor (2025).

- b) Variância: Indica o nível de dispersão das medições diárias de um consumidor em relação à sua média nesse dia. Assim como o valor médio, a variância de um usuário fraudulento

será menor do que a variância de um usuário honesto.

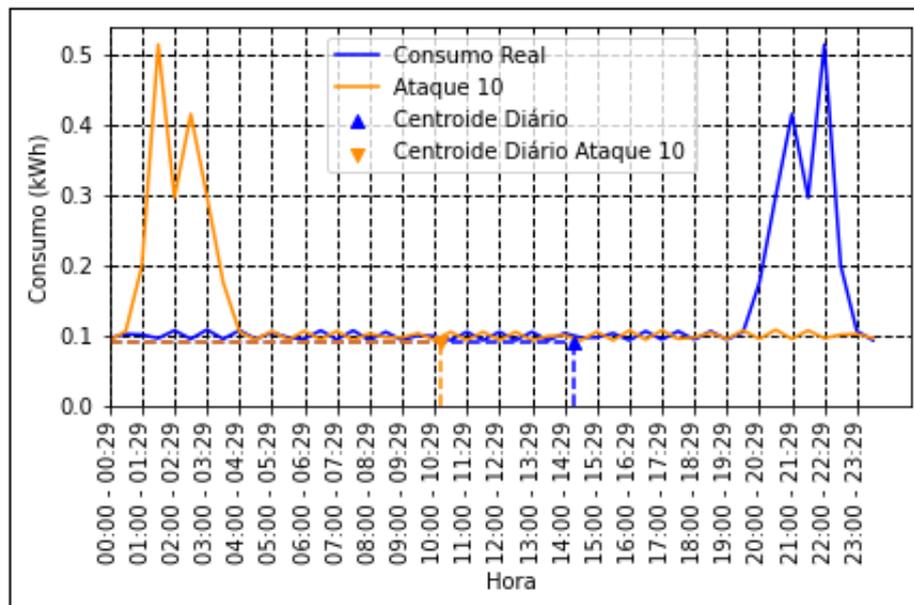
- c) Centróide: Alguns tipos de fraudes podem não ser facilmente detectados apenas com a adição de atributos estatísticos. As coordenadas do centróide, além de fornecer informação adicional sobre o consumo de energia (magnitude), contêm informações sobre a sua distribuição no tempo. Isso pode ajudar a detectar uma PNT, pois alguns ataques tendem a ocorrer em horários específicos do dia, reduzindo o consumo de energia apenas durante aquele intervalo. As coordenadas do centróide \bar{x} e \bar{y} podem ser calculadas da seguinte forma, conforme Strang e Herman (2016):

$$\bar{x} = \frac{\sum_{t=1}^T t\mathbf{x}}{\sum_{t=1}^T \mathbf{x}} \quad (48)$$

$$\bar{y} = \frac{0,5 \sum_{t=1}^T \mathbf{x}^2}{\sum_{t=1}^T \mathbf{x}} \quad (49)$$

Na Figura 25, observa-se o valor da coordenada \bar{x} , que é fundamental neste caso para ajudar a diferenciar o usuário honesto do fraudulento para esse tipo de fraude $f = 10$.

Figura 25 – Consumo real e fraude do tipo $f = 10$ com suas coordenadas dos centróides.



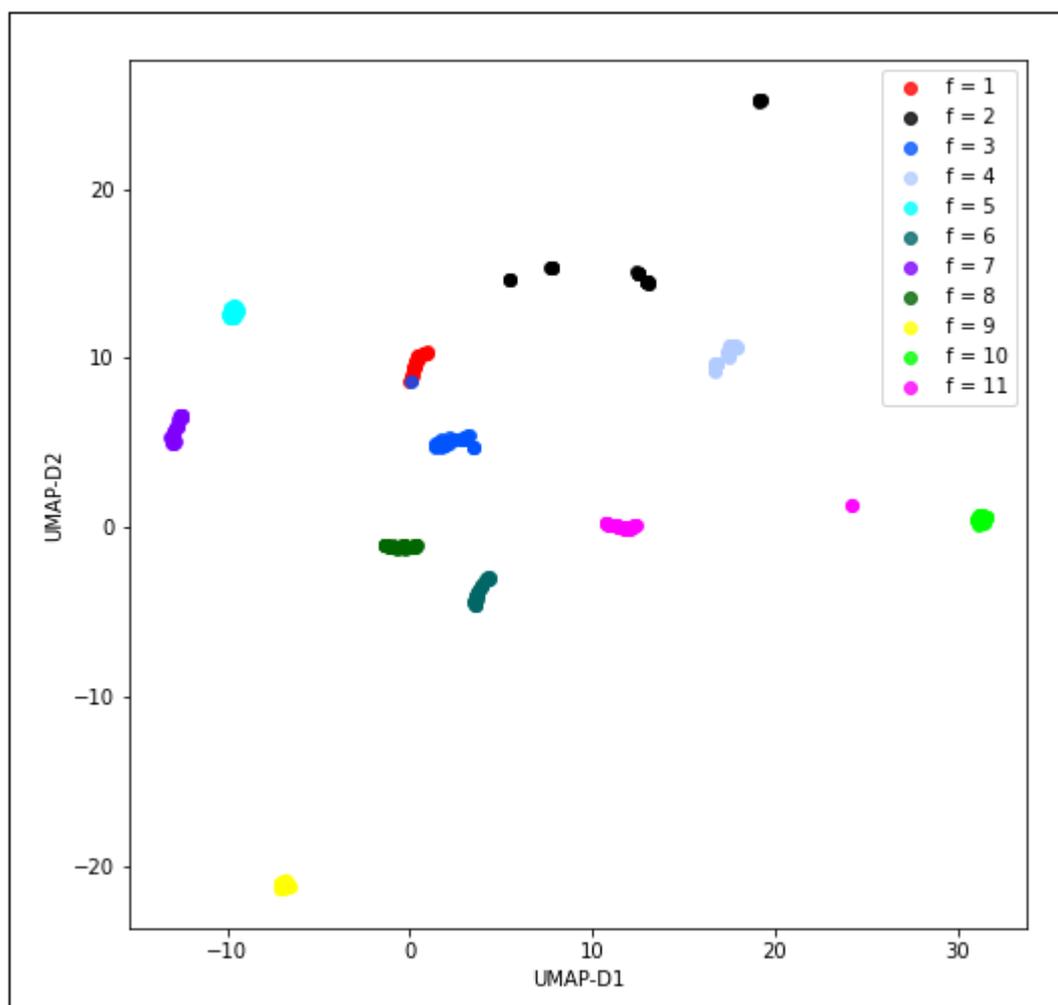
Fonte: O Autor (2025).

- d) UMAP: Realizaram-se vários testes para identificar o valor ideal de τ que melhorasse o desempenho de classificação sem causar um aumento excessivo no tempo de treinamento

dos modelos. No total, foram desenvolvidos oito modelos com valores de τ variando de dois até nove.

Para visualizar como o UMAP separa as classes em duas dimensões, foi escolhido um consumidor aleatoriamente na base de dados, e foram geradas 30 amostras de cada tipo de fraude. A Figura 26 é a redução realizada pelo UMAP, e como cada classe ficou separada em um agrupamento diferente. Há apenas uma leve sobreposição de algumas fraudes do tipo $f = 3$ no agrupamento das fraudes do tipo $f = 1$, o que é esperado, pois esses ataques são bastante semelhantes em sua concepção.

Figura 26 – Redução de dimensionalidade de um consumidor através do UMAP.

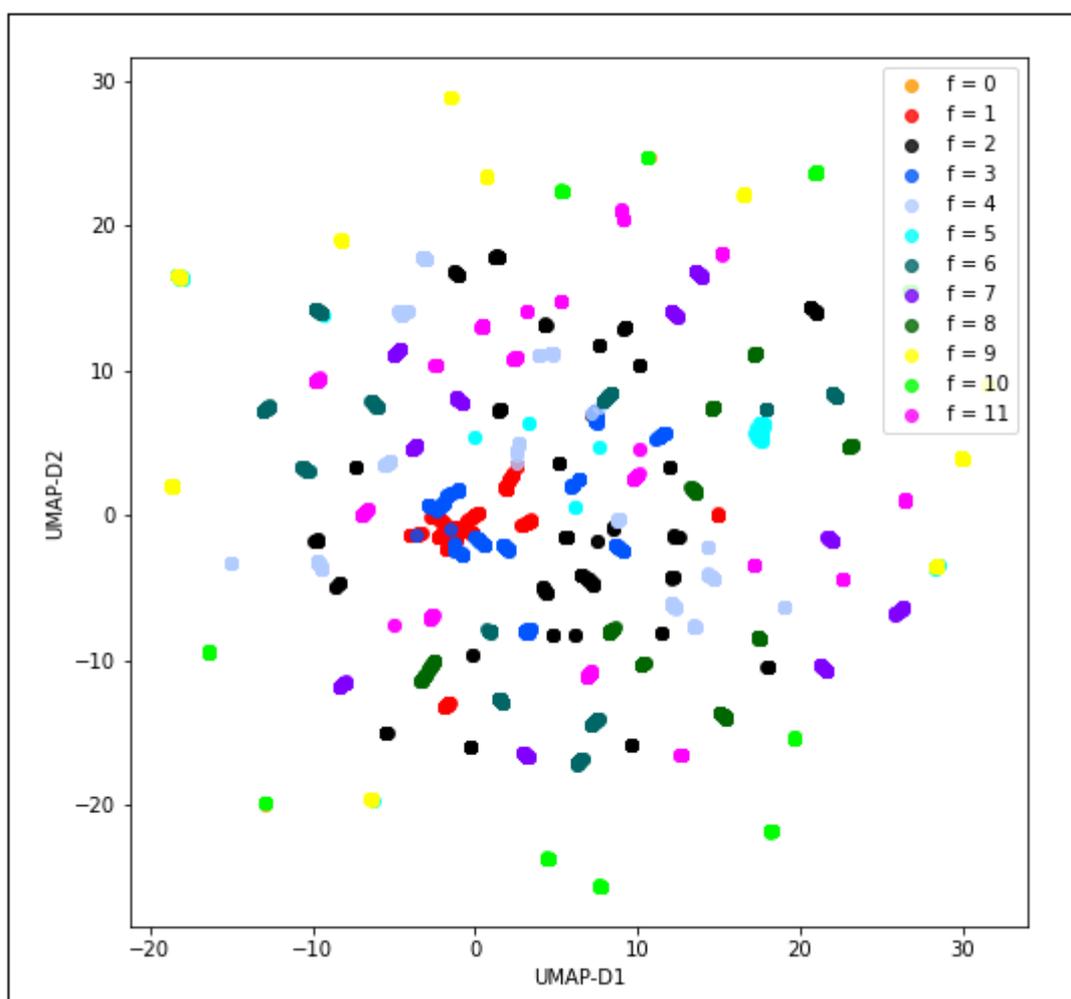


Fonte: O Autor (2025).

Na Figura 27 pode ser visualizado o resultado da redução de dimensionalidade do UMAP para um conjunto de dez consumidores com 30 amostras de cada tipo de fraude. Observa-se novamente a semelhança entre as fraudes do tipo $f = 1$ e $f = 3$, e também ocorre certa

semelhança entre as do tipo $f = 9$ e $f = 10$ em alguns casos. De maneira geral, nota-se uma boa separação entre os grupos, mesmo considerando diferentes consumidores e utilizando apenas duas dimensões. Isso indica que os atributos extraídos pelo UMAP podem ser empregados como entrada para os modelos de IA, especialmente se mais dimensões forem incorporadas no processo de redução de dimensionalidade, uma vez que a separação dos grupos se tornará mais clara.

Figura 27 – Redução de dimensionalidade de dez consumidores através do UMAP.



Fonte: O Autor (2025).

4.4 MODELOS DESENVOLVIDOS

Com o intuito de melhorar o desempenho dos modelos desenvolvidos por Souza *et al.* (2024), neste trabalho foram propostos novos modelos que utilizam os atributos extraídos através do UMAP e séries temporais com uma taxa de amostragem diferente daqueles. Alguns utilizam apenas a própria série temporal como entrada, enquanto outros usam combinações

dessas séries com atributos estatísticos (média e variância), atributos temporais (coordenadas do centroide da série temporal) e/ou atributos extraídos pelo UMAP.

Uma subamostragem do intervalo semi-horário para o intervalo horário foi realizada em todos os modelos inicialmente desenvolvidos em Souza *et al.* (2024). Essa operação teve o objetivo de possibilitar uma comparação mais eficaz dos resultados daquele trabalho com os de outros estudos previamente encontrados na literatura, os quais utilizam a mesma base de dados, como os trabalhos de Jokar, Arianpoo e Leung (2016), Messinis, Rigas e Hatziargyriou (2019) e Chuwa e Wang (2021).

Como as redes LSTM têm sido amplamente aplicadas em diversos problemas de reconhecimento de sequências, como previsão e classificação de séries temporais (Goodfellow; Bengio; Courville, 2016; Pham, 2021), em Souza *et al.* (2024) foi proposto um modelo final (M8) composto por dois módulos (rede modular). O primeiro módulo inclui uma MLP que processa apenas atributos estatísticos e temporais, enquanto o segundo módulo é uma LSTM que processa a própria série temporal. As saídas desses módulos são concatenadas e fornecem a entrada para uma outra MLP, que determina se o usuário cometeu fraude ou não.

Os modelos M1 a M6 são compostos exclusivamente por redes do tipo MLP, enquanto o modelo M7 é baseado em uma rede LSTM. Todos esses modelos foram propostos por Souza *et al.* (2024) e estão resumidos na Tabela 5. Os modelos M1, M2, M3 e M7 possuem 24 entradas, correspondentes à série temporal de consumo de energia horária. O modelo M4 inclui uma entrada adicional, que representa a média do consumo, enquanto o M5 adiciona tanto a média quanto a variância como entradas. O M6, por sua vez, incorpora a média, a variância e as coordenadas \bar{x} e \bar{y} do centroide. Todos os modelos levam em consideração a classe do usuário honesto na saída. O M1 gera nove saídas, sendo uma para o usuário honesto (classe 0) e oito para fraudes que são consideradas facilmente detectáveis (classes 2, 5, 6, 7, 8, 9, 10 e 11). O M2 considera apenas o usuário honesto e os três tipos de ataques considerados mais difíceis de detectar (classes 1, 3 e 4). Já os demais modelos abrangem todas as possibilidades de fraudes. O modelo M8 utiliza uma rede modular, que combina um MLP para processar os atributos extraídos das séries temporais e uma rede LSTM para processar as séries temporais propriamente ditas. As saídas de cada módulo são então concatenadas e fornecidas como entrada para outra rede MLP, que realiza a classificação do usuário.

Tabela 5 – Modelos desenvolvidos por Souza *et al.* (2024).

Modelo	Rede	Entradas	Saídas
M1	MLP	24	9
M2	MLP	24	4
M3	MLP	24	12
M4	MLP	25	12
M5	MLP	26	12
M6	MLP	28	12
M7	LSTM	24	12
M8	Modular	28	12

Fonte: Souza *et al.* (2024).

Como mencionado, o objetivo deste trabalho é aprimorar o desempenho de classificação em comparação com os modelos apresentados na Tabela 5. A abordagem de subamostragem foi descartada, pois ela resulta em perda de informações relevantes, como variações rápidas ou pequenas flutuações entre as medições. Assim, foi desenvolvido o modelo M9, que inclui um módulo MLP, o qual recebe a média, a variância e as coordenadas \bar{x} e \bar{y} do centroide, enquanto o módulo LSTM processa a série temporal com medições semi-horárias de 48 períodos. Os modelos M10 a M17, por sua vez, também utilizam os atributos gerados pelo UMAP, sendo que o M10 utiliza duas dimensões, e a cada modelo subsequente uma nova dimensão é adicionada. Na Tabela 6, estão apresentados todos os novos modelos propostos para este fim.

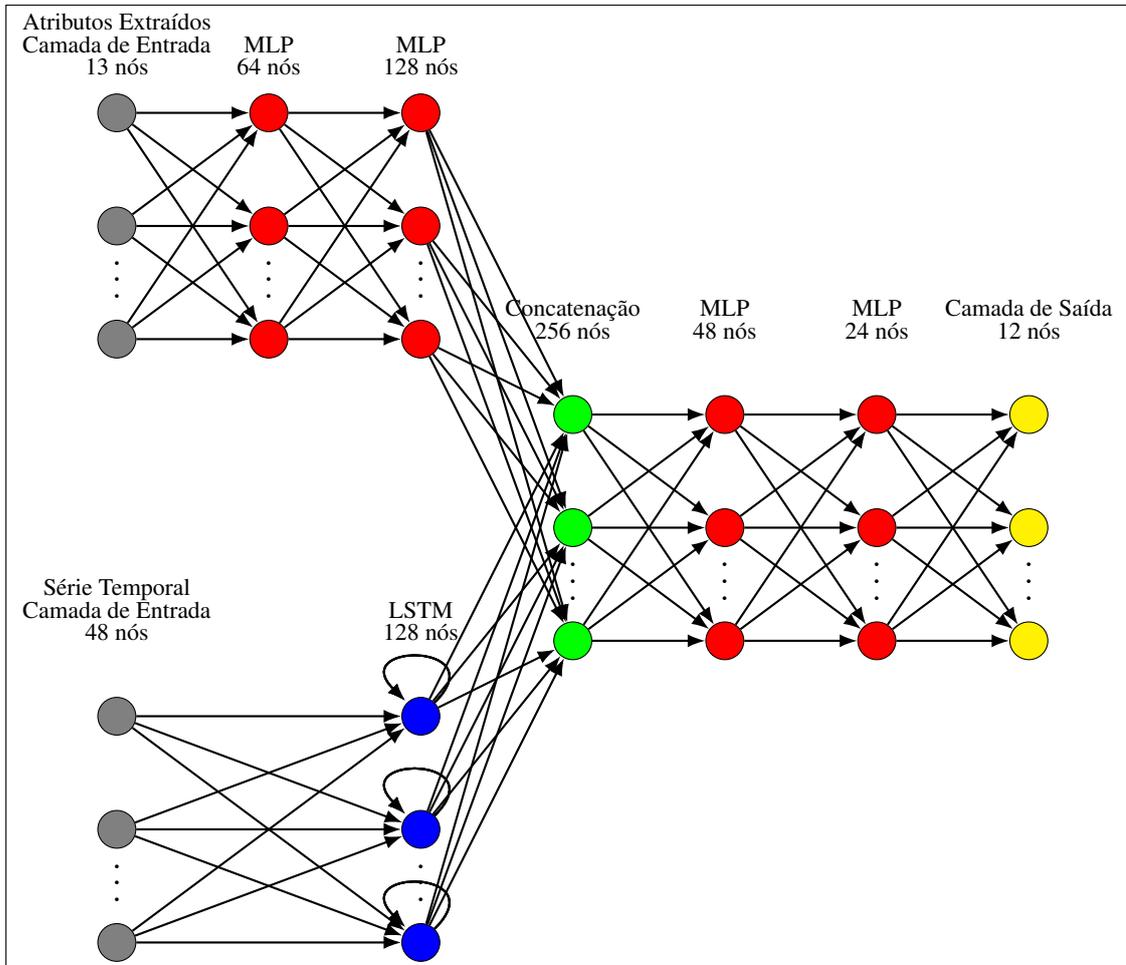
Tabela 6 – Novos modelos propostos.

Modelo	Rede	Entradas	Saídas
M9	Modular	52	12
M10	Modular	54	12
M11	Modular	55	12
M12	Modular	56	12
M13	Modular	57	12
M14	Modular	58	12
M15	Modular	59	12
M16	Modular	60	12
M17	Modular	61	12

Fonte: O Autor (2025).

A Figura 28 é a topologia dessa rede modular proposta para o modelo M17 que contém 61 entradas no total e 12 saídas. Para todos os modelos com redes modulares, se utilizou a função de ativação tangente hiperbólica, ReLU e *softmax*, para a LSTM, MLP e camada de saída, respectivamente.

Figura 28 – Rede modular.



Fonte: Adaptado de Souza *et al.* (2024).

4.5 MÉTRICAS DE AVALIAÇÃO

A matriz de confusão, também conhecida como tabela de contingência (Fawcett, 2006), é uma métrica de avaliação de modelos utilizada em problemas de classificação. Ela serve como base para todas as outras métricas que são comumente utilizadas na avaliação de desempenho de modelos. Essa matriz contém o número de elementos que foram classificados corretamente e incorretamente em cada classe do problema. Generalizando para um caso com N_c classes, ela terá o formato $N_c \times N_c$. Assim, para o caso de classificação binário, ela será uma matriz 2×2 . Essa matriz de confusão binária é dividida em quatro células, sendo que as linhas representam o valor real da classe e as colunas o valor previsto por determinado modelo. As células, são:

- a) Verdadeiro Positivo (*True Positive* - TP): Indica a quantidade de elementos da classe positiva que foram corretamente classificados como da classe positiva;
- b) Verdadeiro Negativo (*True Negative* - TN): Indica a quantidade de elementos da classe negativa que foram corretamente classificados como da classe negativa;
- c) Falso Positivo (FP): Indica a quantidade de elementos da classe negativa que foram erroneamente classificados como da classe positiva. Também é conhecido como erro tipo I;
- d) Falso Negativo (FN): Indica a quantidade de elementos da classe positiva que foram erroneamente classificados como da classe negativa. Também é conhecido como erro tipo II.

A ilustração de uma matriz de confusão para um problema binária pode ser visualizada na Figura 29.

Figura 29 – Matriz de confusão binária.

		Previsto	
		Negativa	Positiva
Real	Negativa	TN	FP
	Positiva	FN	TP

Fonte: O Autor (2025).

No entanto, em problemas multiclases, não é possível determinar diretamente os valores de TP, TN, FP e FN. Para isso, esses valores são calculados para cada classe de forma individual. Em um problema com três classes (C1, C2 e C3), a matriz de confusão pode ser representada de acordo com a Figura 30.

Figura 30 – Matriz de confusão para 3 classes.

		Previsto		
		C1	C2	C3
Real	C1	C_{11}	C_{12}	C_{13}
	C2	C_{21}	C_{22}	C_{23}
	C3	C_{31}	C_{32}	C_{33}

Fonte: O Autor (2025).

Por exemplo, considerando a classe C1, ela será considerada positiva quando for prevista como C1, e negativa quando prevista como qualquer outra classe diferente de C1. Nesse contexto, TP corresponderia ao valor de C_{11} . TN seriam os valores de C_{22} , C_{23} , C_{32} e C_{33} . FP corresponderiam aos valores de C_{21} e C_{31} . FN seriam os valores de C_{12} e C_{13} .

Em posse desses valores da matriz de confusão, podem-se calcular as métricas de avaliação utilizadas neste trabalho: precisão (P), *recall* (R) e *F1-score*. A precisão é a razão entre o valor de TP e a soma de TP com FP. Essa métrica indica, de todos os casos previstos como positivos, quantos eram efetivamente positivos:

$$P = \frac{TP}{TP + FP} \quad (50)$$

O *recall*, também conhecido como sensibilidade, é a razão entre TP e a soma de TP com FN. Ele representa a proporção de casos realmente positivos que foram corretamente identificados pelo modelo. Um valor elevado de *recall* indica que o modelo consegue identificar corretamente a maioria das instâncias positivas, reduzindo a quantidade de falsos negativos:

$$R = \frac{TP}{TP + FN} \quad (51)$$

Por fim, o *F1-score* é definido como a média harmônica entre as duas métricas anteriores. Ele é particularmente útil quando se precisa equilibrar essas duas métricas em uma tarefa de classificação, como no caso da detecção de PNTs, onde classificar um usuário honesto como fraudulento, ou o contrário, acarreta implicações e custos distintos:

$$F1\text{-score} = \frac{2PR}{P + R} \quad (52)$$

4.6 METODOLOGIA DE AVALIAÇÃO

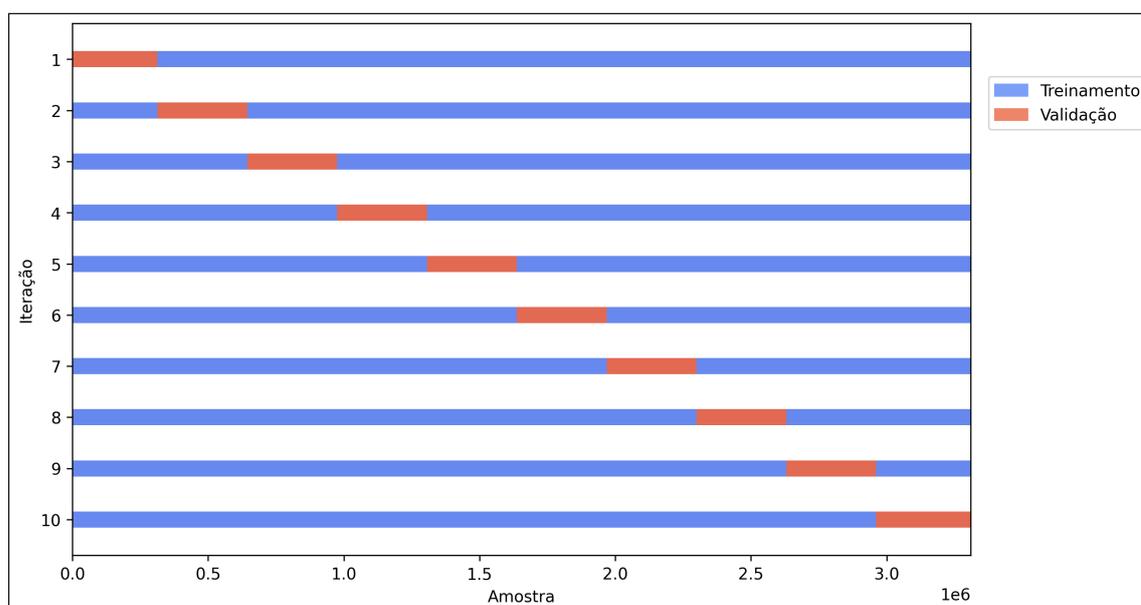
Na Tabela 7 são apresentados os valores das partições dos conjuntos de treinamento, validação e teste da base de dados, definidos para todos os modelos. Além disso, estão indicados o número de épocas de treinamento, a quantidade de épocas do *early stopping* para monitorar a perda de validação, a fim de evitar *overfitting*, e o número de mini-lotes (*batch*).

Tabela 7 – Parâmetros de treinamento.

Treinamento (%)	Validação (%)	Teste (%)	Épocas	<i>Early stopping</i>	<i>Batch</i>
60	20	20	300	50	512

Para o modelo de melhor desempenho, utilizou-se a técnica de validação cruzada do tipo k -fold, com $k = 10$. A Figura 31 ilustra como os conjuntos de treinamento e validação são particionados em cada iteração. Para cada uma delas, o modelo é treinado com $k - 1$ partes dos dados e validado na parte restante. Essa abordagem permite uma estimativa mais robusta e confiável do desempenho do modelo, especialmente no que diz respeito à sua capacidade de generalização.

Figura 31 – Validação cruzada k -fold.



Fonte: O Autor (2025).

Essa técnica foi aplicada exclusivamente ao modelo de melhor desempenho, em razão de seu elevado custo computacional. Conforme pode ser observado na Figura 31, a base de dados utilizada contém mais de 3 milhões de amostras, o que torna inviável, do ponto de vista computacional, repetir o processo de treinamento k vezes para todos os modelos desenvolvidos.

5 DISCUSSÃO DOS RESULTADOS

Os diferentes modelos foram utilizados para demonstrar como a inclusão de cada atributo extraído contribui para a melhoria do processo de classificação do usuário. Salvo indicação em contrário, todos os resultados apresentados neste trabalho correspondem à média de cada métrica para todas as 12 classes consideradas e são referentes a todo o conjunto de teste. Na Tabela 8, os valores de precisão, *recall* e *F1-score* dos modelos treinados exclusivamente para identificar as classes fáceis (M1) e difíceis (M2) podem ser observados. Como esperado, o modelo M1 produziu resultados significativamente superiores aos do modelo M2.

Tabela 8 – Resultados dos modelos M1 e M2.

Modelo	Rede	Entradas	Saídas	<i>P</i>	<i>R</i>	<i>F1-score</i>
M1	MLP	24	9	0,909	0,909	0,909
M2	MLP	24	4	0,633	0,635	0,631

Fonte: Souza *et al.* (2024).

É importante considerar todos os tipos de fraudes para obter um modelo que se ajuste adequadamente à realidade. Por esse motivo, os modelos M3, M4, M5 e M6 foram desenvolvidos. O modelo M3 utiliza exclusivamente a série temporal como entrada, enquanto atributos estatísticos e temporais foram extraídos para aprimorar a detecção de PNTs e fornecidos como entradas adicionais nos modelos M4, M5 e M6. A Tabela 9 apresenta os detalhes desses modelos, demonstrando que, à medida que novos atributos são incorporados, há uma leve melhoria no processo de classificação.

Tabela 9 – Resultados dos modelos M3, M4, M5 e M6.

Modelo	Rede	Entradas	Saídas	<i>P</i>	<i>R</i>	<i>F1-score</i>
M3	MLP	24	12	0,721	0,728	0,723
M4	MLP	25	12	0,747	0,744	0,746
M5	MLP	26	12	0,759	0,766	0,761
M6	MLP	28	12	0,777	0,786	0,779

Fonte: Souza *et al.* (2024).

Uma questão importante a ser considerada ao realizar testes com diferentes modelos de IA é o tempo necessário para treiná-los. Se um modelo se tornar excessivamente complexo (com muitas variáveis e/ou camadas), embora possa haver uma melhoria nas métricas de avaliação, o tempo de treinamento pode se tornar excessivo, tornando-o inviável. A Tabela 10 apresenta o

tempo de treinamento dos modelos M3, M4, M5 e M6. Nota-se que o tempo de treinamento entre esses modelos foi similar, indicando que a inclusão de atributos adicionais não compromete a viabilidade desses modelos nesse aspecto.

Tabela 10 – Tempo de treinamento dos modelos M3, M4, M5 e M6.

Modelo	Rede	Tempo (minutos)
M3	MLP	54,4
M4	MLP	54,6
M5	MLP	54,7
M6	MLP	55,1

Fonte: O Autor (2025).

Esses resultados da Tabela 9 ainda não são adequados para a criação de um modelo prático, pois é crucial que um consumidor desonesto não seja erroneamente classificado como honesto, o que poderia acarretar prejuízos financeiros para a empresa que poderão ser repassados para os demais consumidores. Uma situação ainda mais grave seria classificar um consumidor honesto como desonesto, o que poderia resultar em problemas jurídicos, éticos e de reputação para a concessionária de energia elétrica.

Com o intuito de aprimorar os resultados apresentados anteriormente, o modelo M7 com a rede LSTM foi proposto, visto que esse tipo de rede foi desenvolvido para trabalhar com séries temporais. Baseado no aumento significativo que o modelo M7 apresentou no processo de classificação, o modelo M8 com a rede modular foi desenvolvido, melhorando ainda mais a detecção de PNT. A Tabela 11 contém o resultado de ambos, o qual traz um indicativo importante, haja vista que a rede modular consegue proporcionar um melhor resultado do que utilizar apenas um tipo de rede. A combinação da rede LSTM, que trata a série temporal, com a rede MLP, que lida com os atributos extraídos, mostrou-se mais eficaz na tarefa de classificar todos os tipos de fraudes. O tempo de treinamento do modelo M8 foi levemente superior, levando cerca de 65 minutos para ser treinado, não tornando-o inviável nesse quesito.

Tabela 11 – Resultados dos modelos M7 e M8.

Modelo	Rede	Entradas	Saídas	P	R	F1-score
M7	LSTM	24	12	0,871	0,869	0,869
M8	Modular	28	12	0,885	0,885	0,885

Fonte: Souza *et al.* (2024).

Neste trabalho, a técnica UMAP foi aplicada com o objetivo de melhorar ainda mais os classificadores modulares, uma vez que a redução de dimensionalidade por meio dela conseguiu separar de forma satisfatória todos os tipos de fraudes em diferentes grupos. Além disso, a série temporal foi mantida em sua forma original, com medições semi-horárias, a fim de evitar a perda de informação durante o processo de subamostragem. Podem ser observadas na Tabela 12, as métricas de avaliação de todos os modelos que utilizam os atributos extraídos pelo UMAP, além dos atributos estatísticos e temporais.

Tabela 12 – Resultados dos modelos M9 até M17.

Modelo	Rede	Entradas	Saídas	P	R	F1-score
M9	Modular	52	12	0,915	0,912	0,912
M10	Modular	54	12	0,941	0,939	0,940
M11	Modular	55	12	0,944	0,944	0,944
M12	Modular	56	12	0,949	0,948	0,948
M13	Modular	57	12	0,953	0,953	0,953
M14	Modular	58	12	0,956	0,955	0,955
M15	Modular	59	12	0,959	0,959	0,959
M16	Modular	60	12	0,962	0,962	0,962
M17	Modular	61	12	0,962	0,962	0,962

Fonte: O Autor (2025).

Os resultados apresentado na Tabela 12 revela uma melhoria significativa no processo de classificação ao utilizar a base de dados no formato semi-horário, algo que não havia sido explorado em trabalhos anteriores como os de Jokar, Arianpoo e Leung (2016), Messinis, Rigas e Hatzigargyriou (2019), Chuwa e Wang (2021) e Souza *et al.* (2024). Essa alteração no formato da base de dados, além de reduzir uma etapa no pré-processamento, resultou em um aumento do F1-score de 0,885 para 0,912, o que representa um crescimento de 3,05%. No que se refere aos atributos extraídos pela técnica UMAP, os modelos M16 e M17, que reduziram a dimensionalidade de 48 para oito e nove dimensões, respectivamente, obtiveram um F1-score de 96,2%. Esse valor é 10,7% superior ao melhor modelo utilizando apenas um tipo de rede (modelo M7) e 8,7% superior à rede modular proposta em Souza *et al.* (2024). Apesar de os modelos M16 e M17 utilizarem séries temporais com maior número de medições e mais atributos extraídos pelo UMAP, o tempo médio de treinamento foi apenas 10 minutos superior ao do modelo M8, o que é um acréscimo relativamente baixo considerando esse ganho expressivo de 8,7% no desempenho de classificação. Observa-se que não houve melhoria na detecção de PNT entre os

modelos M16 e M17, apesar do acréscimo de uma dimensão no processo de redução de dimensionalidade entre eles. Portanto, optou-se por utilizar o modelo M16 para realizar a validação cruzada e para a análise do seu desempenho de classificação por classes.

O resultado da validação cruzada para o modelo M16 pode ser visualizado na Tabela 13. Através dos valores de média e desvio padrão de cada iteração, constata-se que esse modelo apresentou grande robustez no processo de classificação.

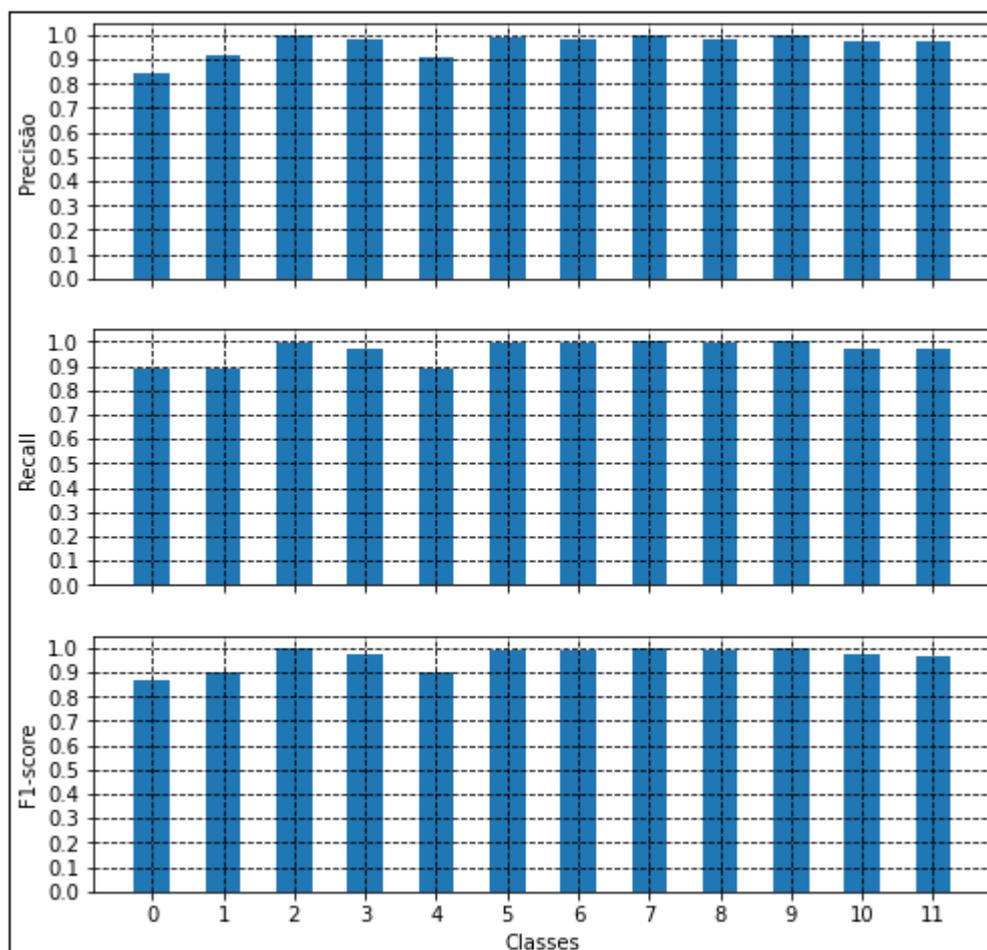
Tabela 13 – Resultados da validação cruzada para o modelo M16.

	<i>P</i>	<i>R</i>	<i>F1-score</i>
Iteração 1	0,963	0,963	0,963
Iteração 2	0,964	0,963	0,963
Iteração 3	0,963	0,963	0,963
Iteração 4	0,963	0,963	0,963
Iteração 5	0,963	0,962	0,963
Iteração 6	0,963	0,963	0,963
Iteração 7	0,962	0,962	0,962
Iteração 8	0,962	0,962	0,962
Iteração 9	0,964	0,963	0,963
Iteração 10	0,963	0,962	0,963
Média	0,963	0,963	0,963
Desvio Padrão	0,0007	0,00055	0,00059

Fonte: O Autor (2025).

O resultado das três métricas para cada classe, referente ao modelo M16, pode ser observado na Figura 32. É possível observar que todas as classes exibiram valores superiores a 0,84 em cada métrica, e em algumas delas o modelo obteve um *F1-score* quase perfeito, atingindo aproximadamente 100%.

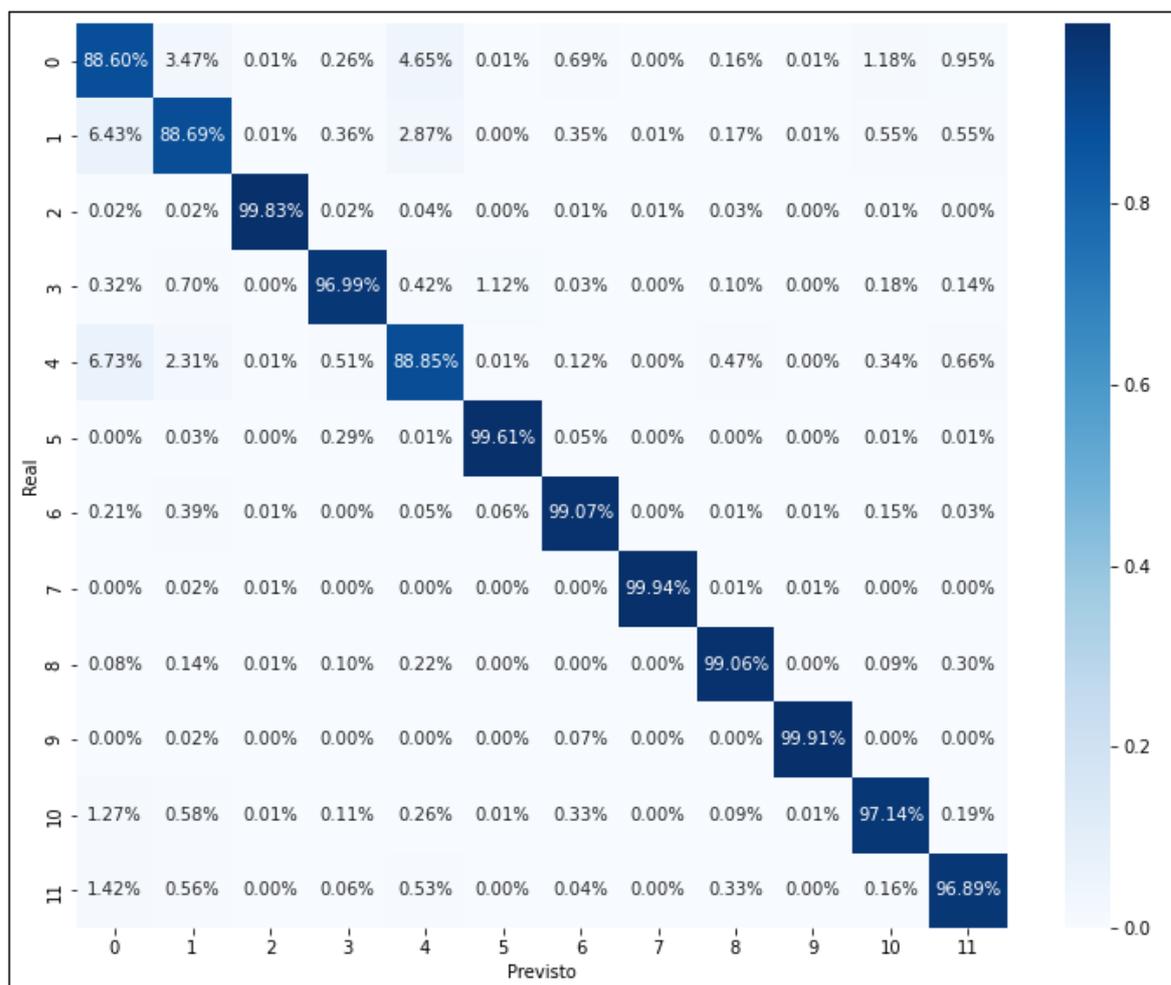
Figura 32 – Métricas por classe para o modelo M16.



Fonte: O Autor (2025).

Para avaliar o desempenho desse modelo de forma mais intuitiva, a Figura 33 é a matriz de confusão para cada classe. O modelo consegue identificar todos os tipos de fraude de maneira eficaz, alcançando uma taxa de detecção superior a 96,89% para as fraudes facilmente detectáveis. Além disso, ele apresenta uma alta taxa de detecção para fraudes mais desafiadoras, com valores superiores a 88,69%, e para o tipo de fraude $f = 3$, o modelo atinge quase 97% de taxa de detecção. Os usuários honestos (classe 0) também são corretamente classificados na maioria das vezes, com uma taxa de 88,60%. Esses resultados representam uma melhora significativa em relação ao melhor modelo proposto por Souza *et al.* (2024), que alcançou uma taxa de detecção para as classes difíceis em torno de 66%.

Figura 33 – Matriz de confusão para o modelo M16.



Fonte: O Autor (2025).

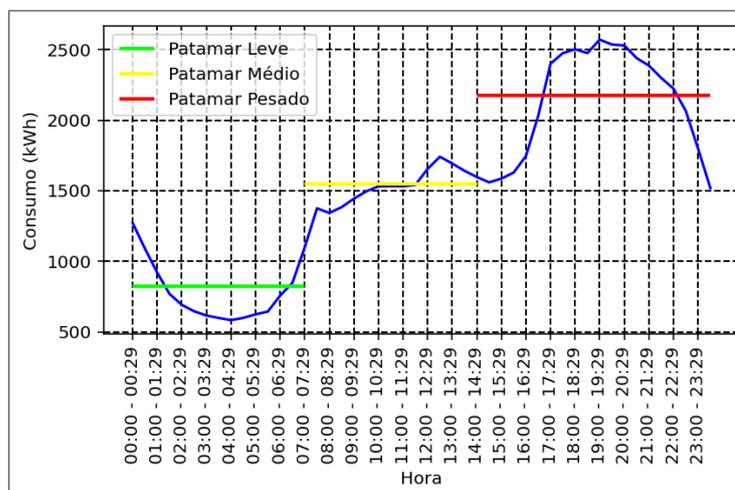
Ao comparar com outros estudos relevantes da literatura, que utilizaram a mesma base de dados e as mesmas métricas de avaliação, o modelo proposto neste trabalho apresentou desempenho superior em diversos aspectos. Por exemplo, no estudo de Chuwa e Wang (2021), que aplicou os mesmos onze modelos de detecção de fraudes e obteve a SVM como o melhor classificador, observou-se um aumento de 21,2% no *F1-score*. Já no *ensemble* apresentado por Bastos *et al.* (2023), que obteve excelente desempenho na classificação, mas considerou apenas seis tipos de fraudes, o *F1-score* foi 3% inferior ao do modelo M16 proposto neste trabalho.

Para verificar a redução do impacto financeiro que o modelo M16 proporcionaria, um estudo em que foram escolhidos aleatoriamente 100 consumidores que não estavam presentes nos conjuntos iniciais de treinamento e teste foi realizado. Para cada um desses consumidores, foram selecionados 30 dias de dados e gerados aleatoriamente os tipos de fraude ao longo desses dias. Essa base gera um total de 6.000 novas amostras.

Na Figura 34, é possível observar o consumo total dos 100 consumidores ao longo do mês,

sem a ocorrência de fraudes. Percebe-se uma nítida divisão do perfil de carga em três patamares: leve, médio e pesado, ao longo das horas.

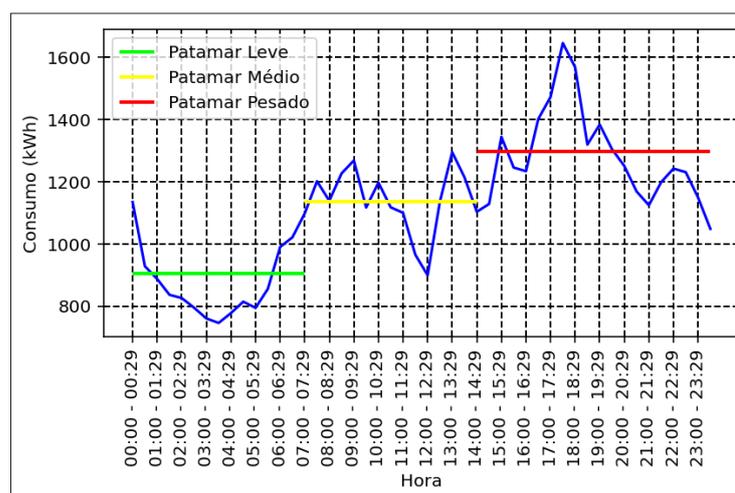
Figura 34 – Consumo total de 100 consumidores sem fraudes ao longo de 30 dias.



Fonte: O Autor (2025).

Já o gráfico da Figura 35 representa o consumo dos mesmos consumidores quando há a ocorrência de fraudes. Observa-se que os patamares de consumo ficam menos definidos devido às fraudes, e o valor de consumo do patamar médio se aproxima do patamar pesado, uma vez que as fraudes tendem a ocorrer nesses períodos. Além disso, nota-se uma redução significativa no consumo de energia.

Figura 35 – Consumo total de 100 consumidores que cometeram algum tipo de fraude ao longo de 30 dias.

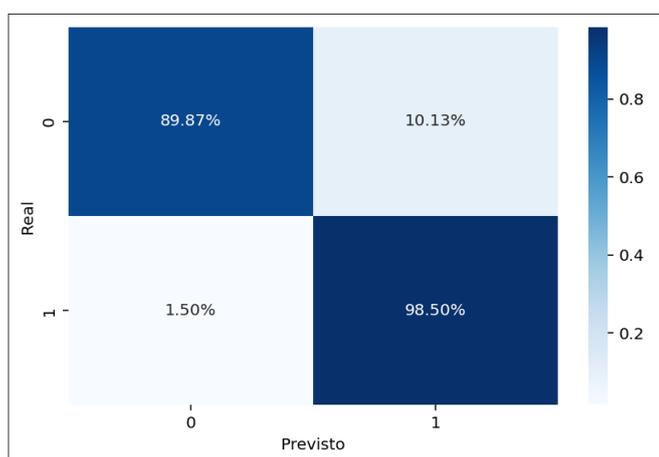


Fonte: O Autor (2025).

O objetivo desta avaliação é analisar os custos que esses consumidores causariam caso tivessem cometido fraude. Para isso, convertem-se as previsões para uma forma binária, em que os consumidores honestos são considerados como classe 0 e as fraudes classe 1. Assim, quando

o consumidor é rotulado e classificado como pertencente a classe positiva, trata-se de um TP, que pode ser contabilizado para a redução de custo. A Figura 36 é a matriz de confusão para essa base de 100 consumidores. O modelo obteve uma taxa de acerto de 98,5% nas amostras em que se cometeram fraude, detectando 2.955 dos 3.000 consumidores.

Figura 36 – Matriz de confusão para modelo M16 referente à base de 100 consumidores.



Fonte: O Autor (2025).

Na Tabela 14 encontram-se as receitas que esses consumidores gerariam para a empresa, tanto em caso de fraude quanto na ausência dela, além do prejuízo causado, considerando uma TME de R\$/kWh 0,257, que corresponde ao valor referente ao ano de 2024 no Estado de Pernambuco.

Tabela 14 – Receitas e prejuízo financeiro causado sem aplicação do modelo M16.

Valor (R\$)	
Receita sem fraude	19.027,73
Receita com todas as fraudes	13.808,66
Prejuízo (sem M16)	5.219,07

Fonte: O Autor (2025).

Isso representa um prejuízo considerável de 27,43% na arrecadação. No entanto, ao utilizar o modelo M16, que detectou 98,5% das amostras positivas, a receita da empresa permaneceria praticamente inalterada, atingindo o valor de R\$ 18.780,92, o que representa uma redução de apenas 1,30% em relação ao cenário sem fraude. Ou seja, ao identificar a fraude, a empresa pode adotar medidas, como interromper o fornecimento de energia e estimar o consumo real antes de emitir a fatura, a fim de evitar perdas financeiras.

Por fim, é importante destacar os casos dos FPs em que não houve fraude, mas o modelo indicou que elas ocorreram. Esse tipo de erro pode levar a ações desnecessárias por parte

da companhia e, embora não gere um impacto financeiro direto relacionado ao consumo de energia, outros custos associados ao seu tratamento, como investigações ou auditorias, podem ser considerados.

6 CONCLUSÕES

O problema das PNT no Brasil e no mundo, e o impacto financeiro que elas causam às concessionárias de energia e conseqüentemente à sociedade foram apresentados. Constatou-se que essas perdas podem ser registradas de diferentes maneiras, todas resultando em prejuízos financeiros e, em alguns casos, podendo causar danos físicos às pessoas, como choques elétricos durante a realização de ligações clandestinas. Dessa forma, para tentar minimizar os malefícios causados pelas PNT, aplicaram-se diferentes modelos de AM para realizar a detecção de um consumidor que cometeu fraude no consumo de energia elétrica. Todos os modelos foram baseados em técnicas de RNA e programados na linguagem *Python*.

A base de dados utilizada foi uma base real com dados de consumo de energia elétrica provenientes de uma AMI de consumidores da Irlanda. Essa base possui dados semi-horários ao longo de quase dois anos. Os consumidores presentes na base são assumidos como honestos, e mediante 11 modelos matemáticos, foram geradas as amostras para simular as fraudes.

Constatou-se que a utilização isolada das séries temporais de consumo de energia elétrica não se mostra suficiente para a construção de modelos de classificação com desempenho adequado às exigências de uma aplicação prática. Por isso, diversos modelos foram desenvolvidos com o objetivo de demonstrar como a inclusão de atributos adicionais pode ser benéfica no processo de classificação de fraude. Foram propostos modelos que incorporam atributos estatísticos (como média e variância) e temporais (coordenadas dos centroides da série temporal), mostrando uma melhoria significativa no processo de classificação.

Com a utilização da técnica UMAP, foi possível extrair automaticamente um maior número de atributos da série temporal, os quais foram incorporados aos modelos, resultando em desempenhos ainda mais satisfatórios na classificação de qualquer tipo de consumidor.

A nova taxa de amostragem da série temporal também demonstrou melhorias na classificação quando comparada àquela empregada em estudos anteriores.

No que diz respeito aos tipos de redes neurais empregadas, a rede recorrente LSTM, que utiliza unicamente a série temporal de consumo como entrada, obteve desempenho superior na tarefa de classificação em comparação às redes MLP previamente adotadas, as quais faziam uso tanto da série temporal como de atributos adicionais. Com base nesse resultado, foi proposta uma arquitetura modular que integra uma rede MLP, responsável pelo processamento dos atributos extraídos, e uma rede LSTM, dedicada ao processamento da série temporal. Essa abordagem modular apresentou altas taxas de detecção tanto para consumidores honestos quanto para

aqueles envolvidos em qualquer tipo de fraude.

Os modelos baseados na arquitetura modular propostos neste trabalho (M10 a M17) demonstram as vantagens da utilização conjunta da série temporal semi-horária e dos atributos extraídos por meio da técnica UMAP. Verificou-se um excelente desempenho de classificação ao empregar esses atributos, com destaque para os modelos M16 e M17, que alcançaram um *F1-score* de 0,962 utilizando oito e nove dimensões, respectivamente. Dentre eles, o modelo M16 se destaca como a opção mais recomendada, por apresentar desempenho equivalente ao M17 com menor complexidade.

Por fim, foi demonstrado como o modelo M16 pode ajudar a reduzir o impacto financeiro causado pelas PNT, resultando em uma diminuição de apenas 1,30% na receita da concessionária, em comparação a um cenário sem fraudes. Esse valor pode ser menor ou maior, a depender da quantidade de consumidores presentes.

6.1 SUGESTÕES DE TRABALHOS FUTUROS

Segue uma breve lista de sugestões para a continuidade da pesquisa:

- a) Aplicar os modelos propostos a diferentes bases de dados, incluindo bases locais, com o objetivo de avaliar seu desempenho em contextos distintos e investigar possíveis aprimoramentos. Entre esses, destaca-se a otimização das arquiteturas por meio de abordagens como busca exaustiva (força bruta) ou algoritmos de otimização mais sofisticados;
- b) Acrescentar outras variáveis de outros domínios, como coeficientes de *Fourier* no domínio da frequência, para tentar obter uma melhor taxa de detecção das fraudes. Espera-se que os valores de magnitude e fase desses coeficientes variem de acordo com o tipo de fraude;
- c) Verificar como os patamares de carga podem ser utilizados como informação adicional para o processo de classificação;
- d) Utilizar técnicas que foram desenvolvidas especificamente para classificação de séries temporais em conjunto com o modelo aqui proposto;
- e) Considerar, além dos consumidores residenciais, outros perfis de consumo, como os comerciais e industriais, por meio do desenvolvimento de um modelo que contemple todos esses segmentos. Tendo em vista que os padrões de consumo variam significativamente

entre esses grupos, a inclusão da informação sobre o tipo de consumidor como variável de entrada pode contribuir para a melhoria na detecção de fraudes, independentemente da categoria analisada.

REFERÊNCIAS

- Agência Nacional de Energia Elétrica. *Perdas de Energia*. 2015. Disponível em: <http://www2.aneel.gov.br/cedoc/aren2015660_Proret_Submod_2_6_V3.pdf>. Acesso em: 22 abr. 2025.
- Agência Nacional de Energia Elétrica. *Perdas de Energia Elétrica na Distribuição*. 2024. Disponível em: <https://git.aneel.gov.br/publico/centralconteudo/-/raw/main/relatorioeindicadores/tarifaeconomico/Relatorio_Perdas_Energia.pdf>. Acesso em: 22 abr. 2025.
- Agência Nacional de Energia Elétrica. *Relatório de Perdas de Energia*. 2025. Disponível em: <<https://portalrelatorios.aneel.gov.br/luznatarifa/perdasenergias>>. Acesso em: 22 abr. 2025.
- ALI, M. *et al.* Timecluster: dimension reduction applied to temporal data for visual analytics. *Vis. Comput.*, Springer-Verlag, Berlin, v. 35, n. 6–8, p. 1013–1026, jun. 2019. ISSN 0178-2789. Disponível em: <<https://doi.org/10.1007/s00371-019-01673-y>>.
- ARANGO, L. G. *Modelagem econômica e regulatória do impacto das perdas comerciais sobre o mercado de energia elétrica*. 2019. 150 f. Tese (Doutorado em Engenharia Elétrica) — Universidade Federal de Itajubá, Itajubá, 2019.
- BARBOSA, M. U. *et al.* Minimização de perdas elétricas em sistemas de distribuição de energia através da alocação ótima de geração fotovoltaica. In: *SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL, 3., 2021, João Pessoa. Anais...* João Pessoa: Galoá, 2021.
- BASTOS, L. *et al.* Data-oriented ensemble predictor based on time series classifiers for fraud detection. *Electric Power Systems Research*, v. 223, p. 109547, 2023. ISSN 0378-7796.
- BC Hydro. *Smart Metering & Infrastructure Program Business Case*. 2011. Disponível em: <<https://www.bchydro.com/content/dam/BCHydro/customer-portal/documents/projects/smart-metering/smi-program-business-case.pdf>>. Acesso em: 22 abr. 2025.
- BENGIO, Y. Practical recommendations for gradient-based training of deep architectures. In: MONTAVON, G.; ORR, G. B.; MÜLLER, K.-R. (Ed.). *Neural Networks: Tricks of the Trade: Second Edition*. Berlin: Springer, 2012. p. 437–478.
- BISHOP, C. M. *Pattern recognition and machine learning*. New York: Springer, 2006. 738 p.
- BRAGA, A. de P.; LUDERMIR, T. B.; CARVALHO, A. C. P. de L. F. *Redes neurais artificiais: teoria e aplicações*. Rio de Janeiro: LTC, 2007. 237 p.
- BURKOV, A. *The hundred-page machine learning book*. [S.l.]: Andriy Burkov, 2019. 160 p.
- CHRISTIANO, L.; VARGAS, P.; ZUBEN, F. V. Redução de perdas em redes de distribuição de energia elétrica através de sistemas classificadores. *Revista Controle & Automação*, v. 14, n. 3, p. 298–308, Setembro 2003.
- CHUWA, M. G.; WANG, F. A review of non-technical loss attack models and detection methods in the smart grid. *Electric Power Systems Research*, v. 199, p. 107415, 2021. ISSN 0378-7796.
- Council of European Energy Regulators. *2nd CEER Report on Power Losses*. 2020. Disponível em: <https://www.ceer.eu/wp-content/uploads/2024/04/C19-EQS-101-03_Report_on_Power_Losses_3.pdf>. Acesso em: 22 abr. 2025.

CYBENKO, G. *Continuous valued neural networks with two hidden layers are sufficient*. Medford: Tufts University: [s.n.], 1988.

CYBENKO, G. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems (MCSS)*, v. 2, n. 4, p. 303–314, Dec 1989.

DAMRICH, S.; HAMPRECHT, F. A. On umap's true loss function. In: RANZATO, M. *et al.* (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2021. v. 34, p. 5798–5809. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/2021/file/2de5d16682c3c35007e4e92982f1a2ba-Paper.pdf>.

DEY, H. S. *et al.* Design and implementation of a novel protection device to prevent tampering and electricity theft in commercial energy meters. *Journal of Computer and Information Technology*, v. 01, n. 01, 2010.

DOMINGUES, I. *et al.* Evaluation of oversampling data balancing techniques in the context of ordinal classification. In: *2018 International Joint Conference on Neural Networks (IJCNN)*. [S.l.: s.n.], 2018. p. 1–8.

FAWCETT, T. An introduction to roc analysis. *Pattern Recognition Letters*, v. 27, n. 8, p. 861–874, 2006.

FERREIRA, H. M. *Uso de ferramentas de aprendizado de máquina para prospecção de perdas comerciais em distribuição de energia elétrica*. 2008. 89 f. Dissertação (Mestrado em Engenharia Elétrica) — Universidade Estadual de Campinas, Campinas, 2008.

GAO, Y.; FOGGO, B.; YU, N. A physically inspired data-driven model for electricity theft detection with smart meter data. *IEEE Transactions on Industrial Informatics*, v. 15, n. 9, p. 5076–5088, 2019.

GAUCHE, E. *Minimização de perdas elétricas em sistemas de distribuição através de reconfiguração de redes*. 1994. 88 f. Dissertação (Mestrado em Engenharia Elétrica) — Universidade Federal de Santa Catarina, Florianópolis, 1994.

GERON, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. 3. ed. Sebastopol: O'Reilly Media, Inc., 2022. 834 p.

GITURA, P. K. *Analysis of factors affecting the quality of power supplied to manufacturers in Kenya*. 2006. 60 p. Dissertação (MBA) — University of Nairobi, Nairobi.

GOODFELLOW, I. J.; BENGIO, Y.; COURVILLE, A. *Deep learning*. Cambridge: MIT Press, 2016. 775 p.

GRAVES, A.; MOHAMED, A.; HINTON, G. Speech recognition with deep recurrent neural networks. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. Vancouver: [s.n.], 2013. p. 6645–6649.

GULARTE, A. P. dos S. *A NON-SUPERVISED ASSET SELECTION FOR PORTFOLIO OPTIMIZATION*. 2023. 104 f. Dissertação (Mestrado em Ciência) — Instituto Tecnológico da Aeronáutica e Universidade Federal de São Paulo, São José dos Campos, 2023.

GUNTURI, S. K.; SARKAR, D. Ensemble machine learning models for the detection of energy theft. *Electric Power Systems Research*, v. 192, n. 2, p. 106904, Mar. 2021.

- HAYKIN, S. *Neural Networks and Learning Machines*. 3. ed. New Jersey: Prentice Hall, 2009. 936 p.
- HEATON, J. *Introduction to neural networks with Java*. Chesterfield: Heaton Research, 2008. 440 p.
- HEBB, D. O. *The organization of behavior*. New York: John Wiley & Sons, 1949. 335 p.
- HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural Computation*, v. 9, n. 8, p. 1735–1780, 1997.
- ISSDA. *The Smart Metering Electricity Customer Behaviour Trials*. 2010. Disponível em: <<https://www.ucd.ie/issda/data/commissionforenergyregulationcer/>>. Acesso em: 22 abr. 2025.
- JANECEK, A. *et al.* On the relationship between feature selection and classification accuracy. In: *Proceedings of the Workshop on New Challenges for Feature Selection in Data Mining and Knowledge Discovery at ECML/PKDD 2008*. Antwerp, Belgium: [s.n.], 2008. (Proceedings of Machine Learning Research, v. 4), p. 90–105.
- JOKAR, P.; ARIANPOO, N.; LEUNG, V. C. M. Electricity theft detection in AMI using customers' consumption patterns. *IEEE Transactions on Smart Grid*, v. 7, n. 1, p. 216–226, 2016.
- KAUFMANN, D.; KRAAY, A.; ZOIDO-LOBATON, P. *Governance matters II: Updated indicators for 2000/01*. Washington, DC: World Bank, 2002. 56 p.
- KHALID, S.; KHALIL, T.; NASREEN, S. A survey of feature selection and feature extraction techniques in machine learning. In: *SCIENCE AND INFORMATION CONFERENCE (SAI), 2014, London. Proceedings... Piscataway: IEEE*. [S.l.: s.n.], 2014. p. 372–378.
- KHAN, Z. A. *et al.* Electricity theft detection using supervised learning techniques on smart meter data. *Sustainability*, v. 12, n. 19, 2020. ISSN 2071-1050.
- LÖNING, M. *et al.* sktime: A unified interface for machine learning with time series. *CoRR*, abs/1909.07872, 2019.
- MAATEN, L. van der; HINTON, G. Visualizing data using t-sne. *Journal of Machine Learning Research*, v. 9, n. 86, p. 2579–2605, 2008.
- MCCULLOCH, W.; PITTS, W. A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, v. 5, p. 127–147, 1943.
- MCDANIEL, P.; MCLAUGHLIN, S. Security and privacy challenges in the smart grid. *IEEE Security Privacy Magazine*, v. 7, n. 3, p. 75–77, 2009.
- MCINNES, L.; HEALY, J.; MELVILLE, J. *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*. 2020. Disponível em: <<https://arxiv.org/abs/1802.03426>>. Acesso em: 22 abr. 2025.
- MESSINIS, G. M.; RIGAS, A. E.; HATZIARGYRIOU, N. D. A hybrid method for non-technical loss detection in smart distribution grids. *IEEE Transactions on Smart Grid*, v. 10, n. 6, p. 6080–6091, 2019.

MÜLLER, A.; GUIDO, S. *Introduction to machine learning with python: A guide for data scientists*. 1. ed. Sebastopol: O'Reilly Media, 2016. 400 p.

NAGI, J. *et al.* Nontechnical loss detection for metered customers in power utility using support vector machines. *IEEE Transactions on Power Delivery*, v. 25, n. 2, p. 1162–1171, 2010.

NARISSETTY, N. N. Chapter 4 - bayesian model selection for high-dimensional data. In: Srinivasa Rao, A. S.; RAO, C. (Ed.). *Principles and Methods for Data Science*. Elsevier, 2020, (Handbook of Statistics, v. 43). p. 207–248. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0169716119300380>>.

NAVANI, J.; SHARMA, N.; SAPRA, S. Technical and non-technical losses in power system and its economic consequence in indian economy. *International Journal of Electronics and Computer Science Engineering*, v. 1, n. 2, p. 757–761, Mar. 2012.

NGAMCHUEN, S.; PIRAK, C. Smart anti-tampering algorithm design for single phase smart meter applied to ami systems. In: *2013 10th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*. [S.l.: s.n.], 2013. p. 1–6.

NIELSEN, A. *Análise Prática de Séries Temporais: Predição com Estatística e Aprendizado de Máquina*. 1. ed. Rio de Janeiro: Starlin Alta Editora e Consultoria Eireli, 2021. 480 p.

NIZAR, A. H. *et al.* Load profiling method in detecting non-technical loss activities in a power utility. In: *2006 IEEE International Power and Energy Conference*. [S.l.: s.n.], 2006. p. 82–87.

Northeast Group LLC. *Electricity Theft and Non-Technical Losses: Global Markets, Solutions and Vendors*. 2017. Disponível em: <<http://www.northeast-group.com/>>. Acesso em: 22 abr. 2025.

Northeast Group LLC. *Electricity Theft and Non-Technical Losses: Quantifying the non-technical loss problem*. 2021. Disponível em: <<http://www.northeast-group.com/>>. Acesso em: 22 abr. 2025.

OLAOLUWA, O. G. Electricity theft and power quality in Nigeria. *International Journal of Engineering Research & Technology*, v. 6, n. 6, p. 1180–1184, June 2017.

PHAM, T. Time–frequency time–space lstm for robust classification of physiological signals. *Scientific Reports*, v. 11, p. 6936, 03 2021.

RAMOS, C. C. *et al.* A novel algorithm for feature selection using harmony search and its application for non-technical losses detection. *Computers & Electrical Engineering*, v. 37, n. 6, p. 886–894, Nov. 2011.

RAMOS, C. C. O. *Caracterização de perdas comerciais em sistemas de energia através de técnicas inteligentes*. 2014. 144 f. Tese (Doutorado em Ciências) — Universidade de São Paulo, São Paulo, 2014.

RASCHKA, S.; MIRJALILI, V. *Python machine learning. 3rd ed.* Birmingham: Packt Publishing, 2019. 770 p.

RODRIGUES, C. E. M. *et al.* Technical loss calculation in distribution grids using equivalent minimum order networks and an iterative power factor correction procedure. *Energies*, v. 14, n. 3, p. 646–668, Jan. 2021.

- ROSENBLATT, F. *The perceptron - A perceiving and recognizing automaton*. Buffalo: Cornell Aeronautical Laboratory, 1957.
- SAEED, M. S. *et al.* Detection of non-technical losses in power utilities—a comprehensive systematic review. *Energies*, v. 13, n. 18, 2020. ISSN 1996-1073.
- SALLAM, A.; MALIK, O. *Electric distribution systems*. New Jersey: John Wiley & Sons, 2011. 522 p. (Hoboken: Wiley-IEEE Press).
- SAVIAN, F. de S. *et al.* Non-technical losses in electricity distribution: a bibliometric analysis. *IEEE Latin America Transactions*, v. 19, n. 3, p. 359–368, June 2021.
- SMITH, T. B. Electricity theft: a comparative analysis. *Energy Policy*, v. 32, n. 18, p. 2067–2076, Dec. 2004.
- SOUZA, M. A. *et al.* Detection of non-technical losses on a smart distribution grid based on artificial intelligence models. *Energies*, v. 17, n. 7, 2024.
- STRANG, G.; HERMAN, E. *Calculus Volume 1*. [S.l.]: OpenStax, 2016. 867 p.
- VIEGAS, J. L. *et al.* Solutions for detection of non-technical losses in the electricity grid: A review. *Renewable and Sustainable Energy Reviews*, v. 80, p. 1256–1268, Dec. 2017.
- WANG, W.; LU, Z. Cyber security in the smart grid: Survey and challenges. *Computer Networks*, v. 57, n. 5, p. 1344–1371, 2013.
- ZANETTA JUNIOR, L. C. *Fundamentos de Sistemas Elétricos de Potência*. 1. ed. São Paulo: Editora Livraria da Física, 2006. 312 p.
- ZANETTI, M. *et al.* A tunable fraud detection system for advanced metering infrastructure using short-lived patterns. *IEEE Transactions on Smart Grid*, v. 10, n. 1, p. 830–840, 2019.
- ZARZÀ, I. de; CURTÒ, J. de; CALAFATE, C. T. Umap for geospatial data visualization. *Procedia Computer Science*, v. 225, p. 1661–1671, 2023. ISSN 1877-0509. 27th International Conference on Knowledge Based and Intelligent Information and Engineering Systems (KES 2023).
- ZHENG, Z. *et al.* Wide and deep convolutional neural networks for electricity-theft detection to secure smart grids. *IEEE Transactions on Industrial Informatics*, v. 14, n. 4, p. 1606–1615, 2018.