

Extração Automática de Características de Cargas em Smart Grid: Geração de Bancos de Dados e Identificação de Novas Cargas – PIIC/UFES

Edital:	Edital PIIC 2020/2021
Grande Área do Conhecimento (CNPq):	Ciências Exatas e da Terra
Área do Conhecimento (CNPq):	Ciência da Computação
Título do Projeto:	Identificação Autônoma de Equipamentos Eletro-Eletrônicos em Smart Grid
Título do Subprojeto:	Extração Automática de Características de Cargas em Smart Grid: Geração de Bancos de Dados e Identificação de Novas Cargas
Professor Orientador:	Leonardo José Silvestre
Estudante:	Gabriel Mota Bromonschenkel Lima

Resumo

Smart Grids são redes elétrica inteligentes, com foco em casas modernas, distribuição inteligente de energia, uso eficiente em indústrias e residências. Casas inteligentes, ao utilizarem agentes inteligentes para identificar dispositivos não inteligentes e seu consumo, muitas vezes, optam por utilizar sensores para cada entrada de energia, uma medida ineficiente, especialmente, devido ao alto custo.

O projeto estuda uma alternativa eficiente para a identificação de eletro-eletrônicos e a automatização da extração de características de cargas. A fundamentação do trabalho foi baseada em pesquisas anteriores realizadas por participantes do projeto, assim como pesquisas externas. A pesquisa foi realizada seguindo um padrão de coleta, tratamento de dados, construção e validação do modelo inteligente gerado.

Foram obtidos resultados que indicam a viabilidade de algumas técnicas de *data augmentation* para dados de corrente elétrica e uma arquitetura de autoencoder mais simples, que é capaz de extrair características, com eficiência, de maneira que modelos mais simples possam atingir elevada precisão em classificação, evitando a produção de um modelo complexo para cada residência.

Para os conjuntos de dados utilizados, o modelo atingiu resultados relevantes, aprendendo a extrair características com apenas um terço dos dados. No caso de uso apresentado, o modelo de autoencoder convolucional apresentou resultados promissores.

Palavras-chave: *Smart Grid, Deep Learning, Autoencoder, Convolutional Neural Network, Data Augmentation, Unstructured Data.*

1 Introdução

Nos últimos anos, podemos observar as Redes Elétricas Inteligentes - *Smart Grids* - crescendo e desempenhando papéis importantes na otimização do consumo de energia, em explicabilidade no uso energético (*insights* sobre o funcionamento dos eletrônicos e seu desempenho), em análises preditivas dentro dos ecossistemas dessas redes (predição de falha em aparelhos eletrônicos, de consumo, de picos e horários mais econômicos) (BATRA et al., 2014; KIM et al., 2017). Ao redor do mundo existem diversas pesquisas e produções acerca dessas tecnologias, indicando uma tendência no aumento da inteligência computacional nessa indústria mundialmente (BATRA et al., 2014; ALI et al., 2020; ZHANG et al., 2018). As *Smart Grids* são sistemas que realizam a distribuição de energia elétrica de forma eficiente, confiável e sustentável, utilizando recursos de tecnologia da informação. Esse padrão de qualidade é aplicável em várias camadas do setor energético, desde a fase de produção do ativo (energia) até a fase de utilização por parte do consumidor. Essas redes elétricas podem ter um grau de inteligência computacional em vários níveis. Elas podem ter inteligência reduzida, apresentando os dados como eles são ao usuário, ou podem ter inteligência elevada, utilizando padrões complexos de serem detectados nos dados para gerar previsões ou classificações.

O projeto trata da utilização de *Smart Grids* em sua fase final, no consumidor, e tem como objetivo a extração automática de características de cargas, assim como seu armazenamento no formato de um banco de cargas, e a detecção de novas cargas na rede elétrica. Para atingir tal feito foi proposto um modelo inteligente, capaz de reduzir o espaço dimensional dos valores das cargas. A redução na dimensionalidade das amostras gera um conjunto menor de valores que representam a mesma carga, aumentando a explicabilidade das cargas e reduzindo os efeitos da “maldição da dimensionalidade” de Bellman (BELLMAN, 1957).

Existem duas formas de se coletar dados para realizar o experimento: coleta em instalações elétricas (residencial ou industrial) e coleta em bancada de laboratório com rede simulada. A coleta em redes residenciais e industriais é excelente para detectar diversos fatores na análise de dados em instalações elétricas inteligentes sem fatores controlados de ambientes simulados, enquanto que a coleta em laboratório fornece um apoio maior quanto à manutenção de configurações de cargas, facilidade e velocidade em descobrir falhas de experimentação.

Diversos dados de eletro-eletrônicos podem ser utilizados em experimentos de redes elétricas. Algumas informações como medições de potência, de corrente elétrica e de tensão elétrica são comumente utilizadas em pesquisas recentes sobre *Smart Grids* (MONACCHI et al., 2014; MAKONIN et al., 2013; KOLTER et al., 2011; BARKER et al., 2012). A coleta de dados do presente projeto foi realizada no Laboratório de Energias Renováveis I, no prédio de pós-graduação em energia da Universidade Federal do Espírito Santo (UFES), campus São Mateus.

Para esse tipo de modelagem, pode-se observar grandes benefícios na utilização de Aprendizado Profundo (AP) - *Deep Learning* (DL) - (AGGARWAL et al., 2018), em que o agente é treinado com uma massa de dados pré-processados e aprende padrões nos dados de treinamento, por meio de um algoritmo conhecido como Rede Neural Artificial (RNA) - *Artificial Neural Network* (ANN) - (AGGARWAL et al., 2018). As RNAs, que são parte do domínio de DL, são algoritmos com muitos parâmetros e camadas que demandam elevada quantidade de dados e tempo para o treinamento. Os agentes inteligentes que podem ser desenvolvidos computacionalmente, por meio de DL, estão situados em um conjunto maior de técnicas de Inteligência Artificial - *Artificial Intelligence* - conhecido como Aprendizado de Máquina (AM) - *Machine Learning* (ML)

(RUSSEL et al., 2002). Desta forma, o projeto aborda o problema de extração de características em cargas por meio de um autoencoder Convolucional Unidimensional - *One-Dimensional Convolutional Autoencoder* (1D-CAE), uma classe muito específica de RNA (CHEN et al., 2020). Veja a ilustração do 1D-CAE utilizada nos experimentos do projeto na figura 1.

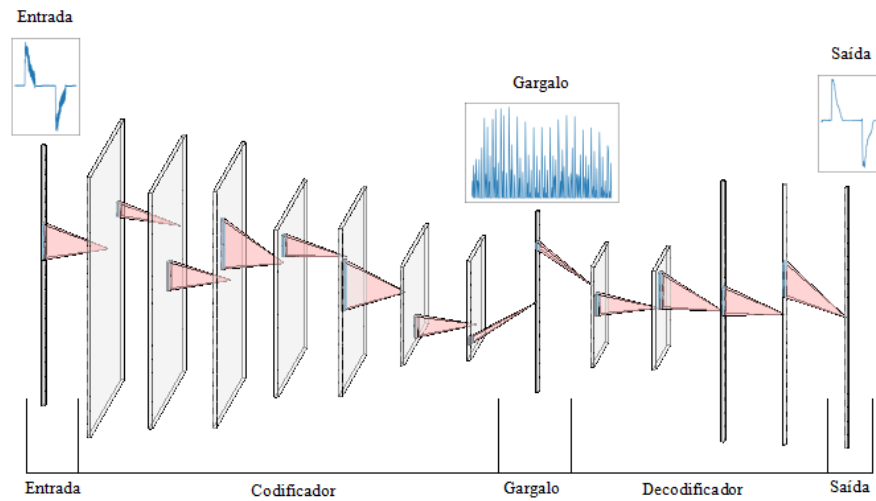


Figura 1: Ilustração do autoencoder produzido no decorrer da pesquisa.

As redes neurais são formadas, em geral, por três blocos estruturais: camada de entrada, camada oculta (que pode conter outras camadas internas) e camada de saída. O componente mínimo de uma rede neural é conhecido como neurônio, ou nó. Cada camada da rede é formada por um agrupamento de nó, cada nó de uma camada interna k da camada oculta se conecta com todos os nós da camada posterior $k+1$ e da camada anterior $k-1$, enquanto que nas camadas de entrada e saída essa interconexão com todos os nós da camada posterior ou anterior ocorre unilateralmente. Os nós são responsáveis por computar um valor, que é a soma ponderada do resultado dos nós anteriores pelo peso de conexão entre nós, em uma função de ativação, para que o resultado dessa função seja utilizado em nós posteriores.

Os autoencoders aprendem valores dos dados de entrada, de maneira não supervisionada, propagando-os até uma camada da estrutura conhecida como gargalo, por sua vez, propaga os valores processados até o final da estrutura (KRAMER, 1991). Na saída do AE os dados processados são comparados com os valores de entrada, onde haverá o cálculo do erro entre o valor processado e os valores de entrada, e então os valores da rede são alterados por meio do algoritmo de Retro-Propagação - *Backpropagation* - (RUMELHART et al., 1986) para que os parâmetros possam realizar transformações mais precisas no próximo ciclo do treinamento ou inferência.

Observando as informações expostas, suponha um sistema de casas em uma *Smart Grid*, que pretendem utilizar uma tecnologia para acompanhar o estado das cargas, assim como o seu consumo. Contudo, é necessário um sistema tão acessível quanto possível. A melhor forma de tornar o sistema acessível financeiramente é evitando o uso de sensores para cada carga dentro da casa. Outra questão muito importante surge quando imagina-se que cada casa possui diferentes eletrodomésticos, que podem estar ligados em diversas tomadas, necessitando de um modelo inteligente específico para cada casa. Esse último problema pode ser contornado com um modelo geral capaz de extrair características das cargas para que um modelo de baixo custo seja implantado em cada casa.

A resposta para as questões levantadas está no presente trabalho, onde informações do estado (ligado ou desligado) das cargas podem ser identificadas por meio de uma única entrada de energia da resultante de todas as cargas - assim como realizado em pesquisas anteriores do projeto (SOARES et al., 2019; FIRMES et al., 2019; PAIXAO et al., 2016) -, respondendo à primeira pergunta, mas não apenas isso, o trabalho realizado ainda propõe um modelo capaz de extrair características das cargas e de gerar dados fáceis de serem processados, devido ao seu tamanho reduzido, diminuindo os custos de modelagem computacional para cada modelo consumível de casa inteligente em termos de tempo, processamento e gerando modelos de classificação mais simples que utilizarão as características extraídas.

No presente projeto, não desejamos classificar nem realizar regressões com os dados de corrente elétrica, o objetivo principal é gerar um modelo capaz de extrair características de cargas para que seja contornado o problema de criar um modelo para cada casa contendo seus próprios eletro-eletrônicos. Com esse modelo, seria possível obter características de qualquer eletro-eletrônico para que as características extraídas pudessem ser utilizadas em um modelo mais simples, reduzindo o tempo de treinamento e o custo de produção do modelo. A criação de um banco de dados de características de cargas também é um dos objetivos do projeto, pois ele representa uma parte importante no procedimento de utilização de um modelo inteligente em escala real, onde o modelo simples é implantado e utilizará os dados de treinamento diretamente desse banco de dados. Outro fator importante sobre o banco de dados é que ele pode armazenar as características e dados de cargas para análises diferentes das utilizadas neste projeto.

2 Objetivos

O objetivo da pesquisa é criar um modelo de IA que possibilite 1) gerar representações, em baixa dimensão, a partir de uma entrada, em alta dimensão, com valores de corrente de um conjunto de cargas em uma Smart Grid, 2) criar um banco de dados com assinaturas de equipamentos eletroeletrônicos, permitindo a busca por equipamentos similares e 3) a identificação de novas cargas em uma rede elétrica.

Objetivos específicos:

- Estudo e análise geral das arquiteturas de redes neurais do tipo AE e CAE, com foco na resolução do problema de extração de características;
- Estudo e análise geral dos seguintes métodos de classificação simples, com foco na classificação das cargas a partir de características extraídas: o K-vizinhos mais próximos – K-Nearest Neighbors (K-NN) - (FIX e HODGES, 1989) e classificação por métricas de distância (similaridade de cossenos);
- Implementação de uma arquitetura de rede neural que seja capaz de gerar representações não ambíguas;
- Montagem de um banco de dados utilizando as representações do modelo implementado.

3 Embasamento Teórico

Em trabalhos anteriores (SOARES et al., 2019; FIRMES et al., 2019; PAIXAO et al., 2016) foram produzidas soluções importantes quanto a classificação de cargas similares. Os modelos produzidos anteriormente apresentaram elevada acurácia na classificação de cargas similares, contudo, a solução poderia ser estendida ainda mais, com um modelo eficiente na simplificação de cargas e armazenamento em bancos de

dados. Os experimentos realizados no presente trabalho tentam resgatar não só a solução de classificação de cargas similares, como também uma forma de facilitação de implantação de modelos residenciais simples que utilizam dados extraídos de um 1D-CAE.

Os AEs são algoritmos muito eficazes na extração de características e podem interpretar características com mais eficiência, em dados complexos, do que algoritmos considerados clássicos, como PCA, t-SNE e LDA (ANOWAR et al., 2021). Os autoencoders estão presentes em diversas arquiteturas de RNAs modernas, devido a sua capacidade de criar *embeddings* - que são características extraídas da entrada de dados - (GOODFELLOW et al., 2014).

O autoencoder utilizado possui algumas camadas com operações muito específicas: amostragem (*pooling* e *sampling*), convolução, planificação (*flattening*), função de ativação ReLU (Unidade Linear Retificadora - *Rectified Linear Unit*) e função de ativação Sigmóide. As três camadas específicas são a camada de codificação, gargalo e decodificação. A ideia principal está na RNA reduzir gradualmente as dimensões da entrada até o menor formato possível, no gargalo, e depois expandir as dimensões até uma saída, cujo resultado deve se aproximar dos valores de entrada. Na camada de codificação a amostragem tem o objetivo de diminuir as dimensões de entrada da camada específica por meio da obtenção dos valores mais significativos da entrada, enquanto que na camada de decodificação a amostragem pega os pontos de saída do gargalo (contendo as características) e expande eles até que atinjam o formato final. A operação de convolução tem o objetivo de dar maior ou menor peso aos valores no interior da rede, por meio de uma operação matricial (LECUN, 1989), desta forma os maiores valores são extraídos pela operação de amostragem e propagados na rede. A função ReLU é responsável por eliminar valores negativos entre camadas, enquanto que a função sigmóide é responsável por estipular os valores finais entre 0 e 1, visto que os dados são normalizados antes de serem inseridos na rede.

Em outras pesquisas, podemos ver abordagens que funcionam bem com modelos em formato em pilha, que é quando utilizamos mais de um modelo em série, com o modelo posterior utilizando as saídas do modelo anterior para realizar inferências. Essa é uma forma de Aprendizado em Conjunto - *Ensemble Learning* - muito utilizada entre as três principais formas de Aprendizado em Conjunto conhecidas, que são *Stacking*, *Bagging* e *Boosting* (ROKACH, 2010). As estratégias de aprendizado citadas são utilizadas com o objetivo de balancear o aprendizado do modelo, por meio da Negociação de Viés e Variância - *Bias-Variance Tradeoff* - (KOHAVI et al., 1996), onde a estratégia de *Stacking* se destaca pela capacidade de reunir modelos diferentes na tentativa de extrair o melhor de cada modelo e melhorar a generalização do modelo final (WOLPERT, 1992).

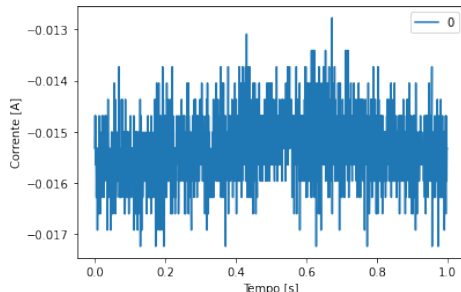
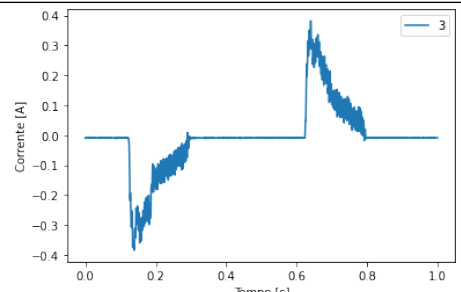
Nas bibliografias sobre *Smart Grids* é muito comum observarmos que as principais técnicas utilizadas são a de séries temporais, como as Redes Neurais Recorrentes - *Recurrent Neural Networks* (RNN) - (AGGARWAL et al., 2018), mais especificamente, as redes de Memória de Longo e Curto Prazo - *Long-Short Term Memory* (LSTM) - (AGGARWAL et al., 2018), que são uma arquitetura moderna capaz de realizar operações de reavaliação de informação para esquecimento ou reforçamento. Com esses modelos, o objetivo, geralmente é ou realizar uma regressão, para prever os valores de consumo dos eletro-eletrônicos, ou realizar uma classificação, para identificar qual é o eletro-eletrônico operando.

4 Metodologia

4.1 Estrutura da Metodologia Adotada

De acordo com a pesquisa bibliográfica e o embasamento para efetuar os experimentos, foi necessário coletar dados de corrente elétrica de algumas disposições diferentes de cargas. Isso ocorre porque os eletro-eletrônicos em um ambiente de análise podem estar conectados em diversas entradas de energia e podem estar ligados ou desligados. Alguns dispositivos são um pouco mais complexos de serem analisados, pois possuem fases de consumo diferentes enquanto estão ligados, como por exemplo, uma geladeira que, enquanto ligada, vai operar baseada em fases intermitentes de congelamento. Um exemplo de amostra de dados coletada segue como na Tabela 1 (Para entender a configuração das cargas observe o ponto 4.3).

Tabela 1 – Exemplo de duas configurações de cargas em bancada que ocorrem entre os dados coletados.

Configuração de cargas	Valor máximo de corrente [A]	Valor mínimo de corrente [A]	Gráfico de corrente por tempo da configuração
Configuração 0	-0,0128	-0,0172	
Configuração 3	0,3817	-0,3840	

Fonte: Produção do(a) próprio(a) autor(a).

O ambiente para a coleta dos dados foi o Laboratório de Energias Renováveis I, que dispunha de uma bancada de medição, monitorada computacionalmente, com quatro entradas para eletro-eletrônicos. Nessa bancada alguns dispositivos eletro-eletrônicos foram utilizados em configurações diferentes (e.g. 4 lâmpadas, 3 lâmpadas e uma entrada vazia, 3 lâmpadas e uma entrada vazia em entradas diferentes, 4 eletrodomésticos distintos, entre outros). Um exemplo ilustrativo da bancada, com a fonte elétrica e o computador de monitoramento está presente na figura 1.

Para realizar os experimentos, foram utilizados um ambiente em nuvem e um computador portátil local. O experimento foi realizado em etapas de 1) análise descritiva dos dados, 2) engenharia de características, 3) modelagem de aprendizado profundo e 4) validação do modelo de inteligência artificial.

No processo de análise descritiva, são analisados os valores nos dados da forma que eles estão, sem qualquer busca específica, analisando apenas qualidades estatísticas nos dados, como média, distribuições de probabilidade, frequência, máximos e mínimos. Com a análise descritiva podemos ter uma noção de como os dados estão e as transformações necessárias. Na engenharia de características os dados precisam ser limpos, normalizados e balanceados. Na etapa de engenharia de características, decidimos também a necessidade de selecionar determinadas colunas de características, em detrimento de outras, e gerar novas características, caso

necessário. Na modelagem de aprendizado profundo é onde o algoritmo de DL selecionado passa pela etapa de teste e estruturação, além de passar por um refinamento de hiperparâmetros como quantidade de camadas, quantidade de neurônios, funções de ativação, *Drop Out*, Normalização de Lote - *Batch Normalization* -, tipo de camada/neurônio (convolucional, residual, recorrente ou comum). Na validação ocorre a análise de desempenho cognitivo do algoritmo, levando em consideração algumas métricas.

4.2 Ambiente e Recursos Utilizados

Os ambientes utilizados foram o Google Colab, que disponibiliza os recursos de acordo com a demanda dos usuários do sistema (no caso deste projeto, 12.72GB de RAM e memória não volátil de até 68GB, CPU Intel® Xeon® de 2.30GHz, GPU NVIDIA Tesla K80), e o um computador pessoal portátil (modelo Lenovo S145, CPU Ryzen 3-3600U de 2.60 GHz, 645GB de SSD, 8GB de RAM DDR4 e GPU Vega 3) com o ambiente do Anaconda 3.

4.3 Coleta de Dados

Foram coletados dados com o objetivo de formar 3 conjuntos de arquivos de Valores Separados por Vírgula - *Comma-Separated Values* (CSV). Os dados coletados são referentes a dezesseis configurações diferentes na bancada de testes (a bancada possui quatro entradas para eletro-eletrônicos), que é a quantidade suportada pelo software de coleta. Desta forma, cada conjunto possui dezesseis classes de dados.

A primeira base já havia sido coletada em trabalhos anteriores (SOARES et al., 2019; FIRMES et al., 2019) e é um conjunto com dados de cargas similares (quatro lâmpadas do mesmo tipo e fabricante). Os dados são de dezesseis configurações possíveis com as quatro lâmpadas (e.g. apenas uma lâmpada conectada na primeira posição, duas lâmpadas conectadas na primeira e terceira posição), assim, podendo ser representadas as configurações pelos números binários de 0 à 15 (e.g. 0000, nenhuma lâmpada conectada; 0101, uma lâmpada no terceiro bocal e uma no primeiro bocal). As duas outras bases, são de eletro-eletrônicos distintos e conectados de diversas maneiras. Assim, a massa de dados ficou no seguinte formato, conforme a Tabela 2.

Tabela 2 – Eletro-eletrônicos utilizados na coleta de dados, cada base possui 16 configurações de cargas dispostas em bancada.

Configuração	Base de cargas similares (Base S1)	Base de cargas distintas 1 (Base D1)	Base de cargas distintas 2 (Base D2)
0	-	lâmpada 1 (1)	ferro elétrico (3)
1	lâmpada (1)	lâmpada 1 (3)	espremedor (4)
2	lâmpada (2)	lâmpada 2 (1)	espremedor (3)
3	lâmpada (1) lâmpada (2)	lâmpada 3 (1)	liquidificador (3)
4	lâmpada (3)	roteador 1 (4)	misteira (2)
5	lâmpada (1) lâmpada (3)	roteador 2 (4)	misteira (1)
6	lâmpada (2) lâmpada (3)	roteador 2 (2)	celular 1 (2) lâmpada 2 (4)
7	lâmpada (1) lâmpada (2) lâmpada (3)	celular 1 (3)	roteador 1 (1) celular 1 (2) lâmpada 2 (4)
8	lâmpada (4)	celular 2 (3)	roteador 1 (1) celular 1 (2)

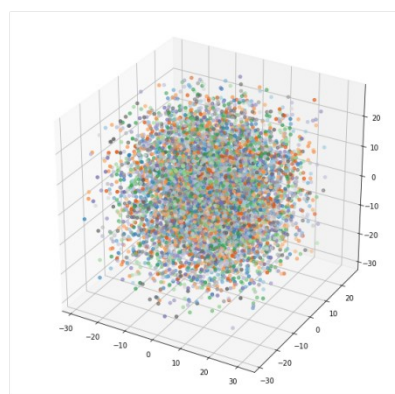
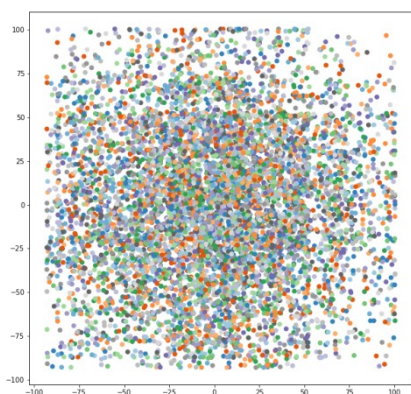
			lâmpada 3 (3)
			lâmpada 2 (4)
			roteador 1 (1)
9	lâmpada (1)	luminária (2)	celular 1 (2)
	lâmpada (4)		lâmpada 3 (3)
			misteira (4)
10	lâmpada (2)	ventilador - velocidade 1	notebook 2 (2)
	lâmpada (4)	(2)	lâmpada 3 (3)
11	lâmpada (1)	ventilador - velocidade 2	espremedor (1)
	lâmpada (2)	(2)	celular 1 (2)
	lâmpada (4)		
12	lâmpada (3)	ventilador - velocidade 3	ferro elétrico (2)
	lâmpada (4)	(2)	misteira (4)
13	lâmpada (1)	cortador de cabelo (3)	roteador 1 (1)
	lâmpada (3)		roteador 2 (2)
	lâmpada (4)		celular 1 (3)
14	lâmpada (2)	notebook 1 (2)	roteador 1 (1)
	lâmpada (3)		roteador 2 (2)
	lâmpada (4)		celular 1 (3)
			lâmpada 2 (4)
15	lâmpada (1)	notebook 2 (2)	notebook 2 (1)
	lâmpada (2)		roteador 2 (2)
	lâmpada (3)		celular 1 (3)
	lâmpada (4)		lâmpada 2 (4)

Fonte: Produção do(a) próprio(a) autor(a).

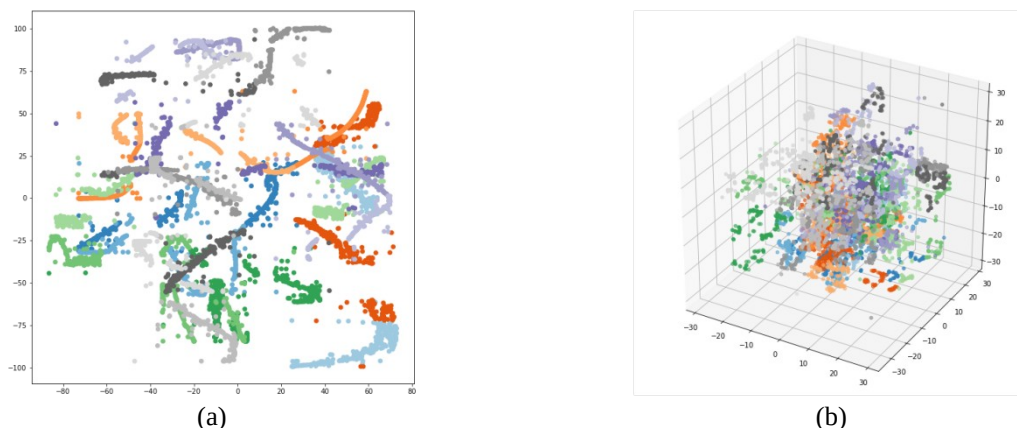
4.4 Análise Descritiva

Na análise descritiva dos dados, é necessário apresentar as características intrínsecas dos dados, da maneira como eles foram coletados. Para projetos desse tipo, em que os dados não são estruturados e, portanto, não possuem correção entre colunas e tabelas, os dados precisam ser apresentados de maneira mais explicável, podendo ser mais facilmente entendidos por seres humanos.

Para uma análise individual das amostras, os valores foram plotados graficamente com ferramentas apropriadas do Python (conforme figuras da tabela 1). Assim, podemos entender a amplitude dos sinais elétricos, a frequência do sinal elétrico e o seu valor médio. Para análise conjunta dos dados pelos grupos que eles são representados foi utilizada uma técnica, especializada em observação de dados em elevadas dimensões por redução estatística, conhecida como t-SNE. A partir dessa técnica, podemos observar a disposição espacial desses dados com 1666 dimensões em um plano de duas dimensões apenas, facilitando o entendimento dos dados e a similaridade entre as classes existentes. Observe as figuras 2 e 3.



(a) (b)
Figura 2: Representação dos dados de cargas similares em (a) duas dimensões e em (b) três dimensões, por meio do algoritmo t-SNE. Cada cor corresponde a uma das classes (configurações) e os pontos possuem coordenadas de acordo com os dados de dimensão reduzida gerados pelo t-SNE.

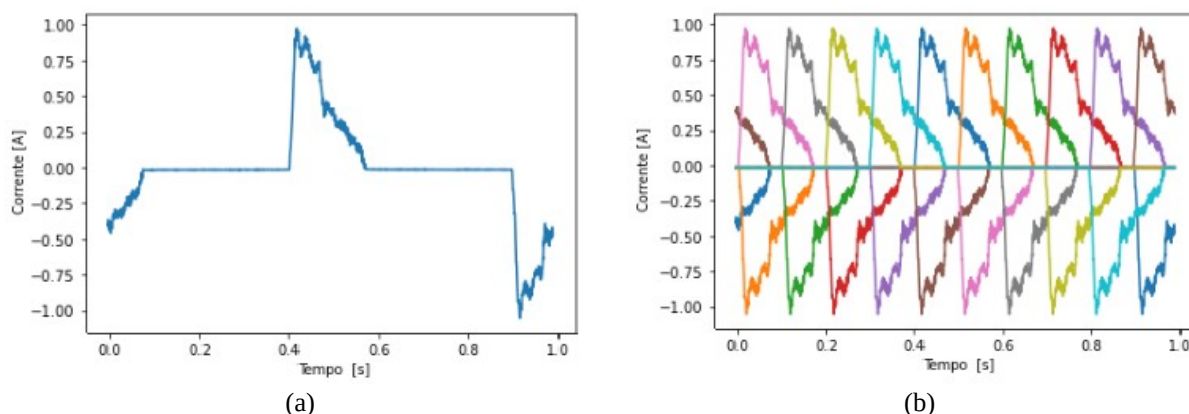


(a) (b)
Figura 3: Representação dos dados de cargas distintas 1 em (a) duas dimensões e em (b) três dimensões, por meio do algoritmo t-SNE. Cada cor corresponde a uma das classes (configurações) e os pontos possuem coordenadas de acordo com os dados de dimensão reduzida gerados pelo t-SNE.

4.5 Engenharia de Características

Para que o modelo possa consumir os dados, esses dados precisam ser adequados, por meio de transformações lineares. Alguns modelos de inteligência artificial realizam operações que necessitam certos padrões existentes nos dados, como por exemplo, ausência de valores nulos, normalização dos dados, tamanhos iguais de entrada, valores numéricos.

Nesta etapa podemos realizar modificações coerentes nos dados de maneira a enriquecê-los, aumentando quantidade ou qualidade, ou os dois. Em um caso específico, o das cargas similares, foi realizado o *shifting* nos dados, como forma de *data augmentation*. O *shifting* é uma operação de deslocar os dados, ou defasá-los, para obter dados do mesmo tipo mas com uma forma diferente do original. Observe a Figura 4.



(a) (b)
Figura 4: (a) Representação de dados de corrente elétrica de uma carga sem qualquer alteração. (b) Representação de dados de corrente elétrica de uma carga elétrica e as novas nove representações por meio de *shifting*.

4.6 Modelagem de Aprendizado Profundo

A principal ideia quando queremos reduzir a dimensionalidade de uma amostra é encontrar uma representação da mesma amostra, porém, utilizando uma quantidade menor de valores. Realizar uma redução na quantidade de características de um conjunto de dados traz alguns benefícios, como simplificação dos dados, redução de redundâncias, aumento na performance de operações e otimização do consumo de memória.

Existem alguns algoritmos considerados eficientes quando se trata de um problema de extração de características (ANOWAR et al., 2021). Contudo, nem todos eles conseguem lidar com dados não estruturados, de forma que sejam analisados padrões com complexidade e abstração tão elevadas como problemas de reconhecimento facial, detecção de padrões em imagens naturais, e os dados de corrente elétrica se enquadram nesse grupo de dados não estruturados. Observe o 1D-CAE modelado, conforme a Tabela 3.

Tabela 3 – Representação das camadas da rede neural 1D-CAE criada.

Nome da camada	Tipo da camada	Dimensão do filtro	Função de ativação	Dimensão de saída	nº de parâmetros	nº de filtros
(a) conv1d	convolução	5	ReLU	1666	1200	200
(b) max_pooling1d	amostragem	2	-	833	-	1
(c) conv1d_1	convolução	4	ReLU	833	80100	100
(d) max_pooling1d_1	amostragem	7	-	119	-	1
(e) conv1d_2	convolução	3	ReLU	119	15050	50
(f) max_pooling1d_2	amostragem	7	-	17	-	1
(g) conv1d_3	convolução	3	ReLU	17	1510	10
(h) flatten	planificação	-	-	170	-	1
(i) reshape	redimensionamento	-	-	17	-	10
(j) conv1d_4	convolução	3	ReLU	17	310	10
(k) up_sampling1d	amostragem	98	-	1666	-	1
(l) conv1d_5	convolução	5	ReLU	1666	51	1
(m) conv1d_6	convolução	5	Sigmóide	1666	6	1

Fonte: Produção do(a) próprio(a) autor(a).

4.7 Avaliação do Modelo de Inteligência Artificial

A avaliação do modelo ocorre em duas etapas. O modelo utilizado é do tipo AE, cujo algoritmo comprime a dimensão dos valores de entrada e depois expande a dimensão dos valores comprimidos com o objetivo de reconstruir os valores de entrada a partir de uma quantidade de valores muito menor do que o da amostra de entrada. Para isso, a primeira avaliação que ocorre é por meio da função de custo de entropia cruzada binária - a dedução pode ser encontrada em “Machine Learning: A Probabilistic Perspective”, por MURPHY (2012) -, para sabermos matematicamente o quanto a saída está parecida com a entrada. Conforme essa função de custo, quanto menor o valor, mais parecida com a entrada a saída está, logo, podemos inferir que os valores extraídos no interior da rede possuem qualidade suficiente para representarem a amostra de entrada, além de poderem ser utilizados para reconstruir essa entrada dada na RNA.

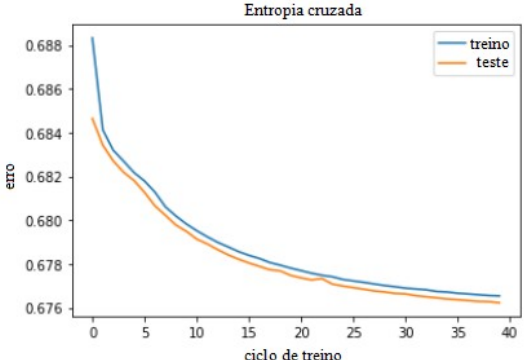
A segunda avaliação ocorre quando tentamos classificar os dados de acordo com suas classes utilizando o K-NN e a similaridade de cossenos. Nessa avaliação, quanto maior a acurácia, precisão, revocação e métrica F1- métricas abordadas precisamente por GÉRON (2019) -, do K-NN sobre os dados extraídos, melhor a

qualidade desses dados para que possam ser utilizados para inferir a classe dos dados originais, contudo, utilizando bem menos valores do que se fossem utilizados os dados originais (9,8 vezes menos valores, redução de 1666 para 170). Para observar de maneira geral os resultados foi utilizada a curva ROC, por meio das curvas de macro-média e micro-média da precisão - uma explicação precisa sobre as métricas pode ser encontrada em publicação de FERRI et al. (2003). A macro-média é calculada por meio da média aritmética da métrica de precisão para cada classe, enquanto que a micro-média é calculada dividindo a soma das amostras verdadeiras positivas pela soma de todas as amostras. Na curva ROC, esses valores são equivalentes à área abaixo da curva.

5 Resultados e Discussão

O modelo autoencoder produzido, conforme a tabela 3, foi treinado apenas utilizando os dados de cargas similares, que representam a terça parte do total de dados de cargas coletadas. Os dados de configurações com cargas distintas foram utilizados para validar o quão eficiente foi a extração de características do autoencoder. Desta forma, a primeira avaliação do modelo ocorreu por meio do erro de entropia cruzada binária, tendo sido analisada a coerência do modelo na reconstrução da entrada, conforme a Tabela 4 e a Figura 1. O modelo indicou estar reconstruindo corretamente, até mesmo sem muitos ruídos, os dados de entrada, permitindo seguir para os testes seguintes.

Tabela 4 – Resultados do critério (função de custo de entropia binária cruzada) no treinamento e validação da rede neural 1D-CAE.

Gráfico de treinamento	Erro de entropia cruzada binária final (treino)	Erro de entropia cruzada binária final (validação)
	0,6767	0,6762

Fonte: Produção do(a) próprio(a) autor(a).

O próximo teste teve como objetivo, avaliar a utilização de *shifting* - uma operação de *data augmentation* com o objetivo de criar dados defasados em relação ao dado original -, conforme a figura 4. Para essa análise, foi utilizada a similaridade de cossenos para observar o quanto a similaridade entre as características extraídas aumentava e a ambiguidade entre *embeddings* de classes distintas diminuía. Na primeira parte do experimento, foi utilizada uma das *embeddings* de cada classe de configuração de cargas similares (Base S1, Tabela 2) do 1D-CAE para classificar as demais com a similaridade de cossenos, sem o recurso de *shifting*, tendo os valores computados, a acurácia foi de 28%. A acurácia da classificação, com a similaridade de cossenos, atingiu um valor de 43% após o treino da rede com o recurso de *shifting*.

Os testes de classificação com o K-NN foram realizados com as duas bases de dados de configurações com cargas distintas, conforme a Tabela 2. O modelo de Autoencoder foi treinado apenas para extrair

características de cargas similares, uma tarefa considerada de elevada complexidade. Então, caso a rede fosse capaz de extrair características, com qualidade, de dados similares, havia a hipótese de a mesma rede conseguir realizar esse feito em cargas distintas. Para computar o quão bem a rede extraia as características, essas características foram utilizadas para classificar as cargas em um modelo mais simples, o K-NN, treinado com apenas 10 amostras de cada classe. Durante o treinamento, o classificador treinado com as características extraídas do 1D-CAE foi capaz de pontuar 88% de acurácia com cargas similares (Tabela 5 e Figura 5). Por outro lado, quando a mesma rede foi testada nas bases de cargas distintas, ela alcançou para a primeira e a segunda base, respectivamente, 98% e 92% de acurácia. Observe as tabelas 6 e 7, assim como as figuras 6 e 7.

Tabela 5 – Experimento 3: Resultados dos critérios (precisão, revocação e métrica F1) de teste das *embeddings* da base de cargas similares (Base S1, Tabela 2) do 1D-CAE com o KNN, sem o recurso de *shifting*.

Configuração das cargas	Precisão	Revocação	F1	Quantidade de amostras
0	1,00	1,00	1,00	122
1	0,91	0,75	0,82	125
2	0,80	0,75	0,78	129
3	1,00	1,00	1,00	115
4	0,92	1,00	0,96	108
5	1,00	1,00	1,00	122
6	0,98	1,00	0,99	121
7	0,82	0,80	0,81	137
8	0,52	0,58	0,55	112
9	1,00	1,00	1,00	114
10	0,68	0,65	0,66	124
11	0,83	0,90	0,87	114
12	0,79	0,77	0,78	115
13	1,00	1,00	1,00	127
14	1,00	0,99	0,99	108
15	0,95	1,00	0,97	127

Fonte: Produção do(a) próprio(a) autor(a).

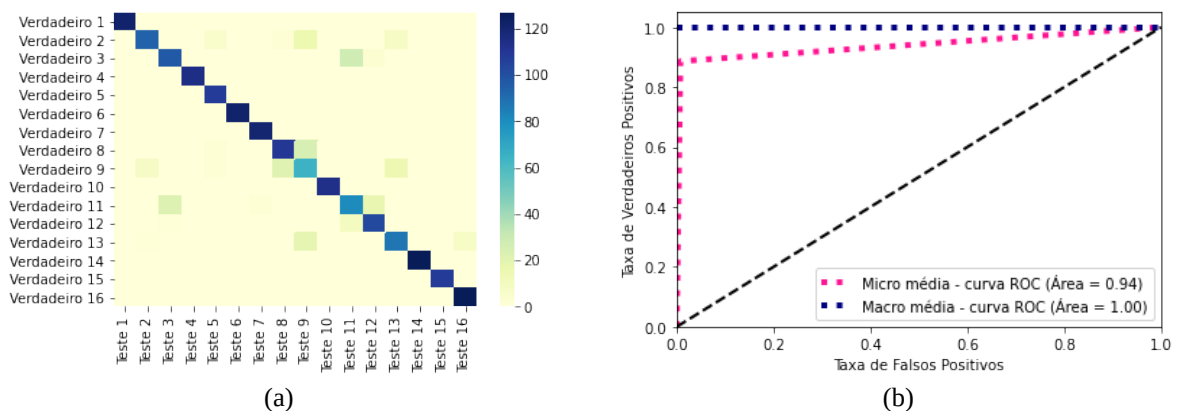


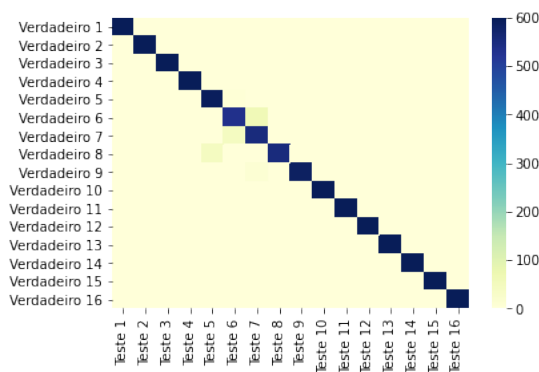
Figura 5: (a) Matriz de confusão do experimento 3. (b) Curva ROC do experimento 3.

Tabela 6 – Experimento 4: Resultados dos critérios (precisão, revocação e métrica F1) de teste das *embeddings* da base de cargas distintas 1 (Base D1, Tabela 2) do 1D-CAE com o KNN, sem o recurso de *shifting*.

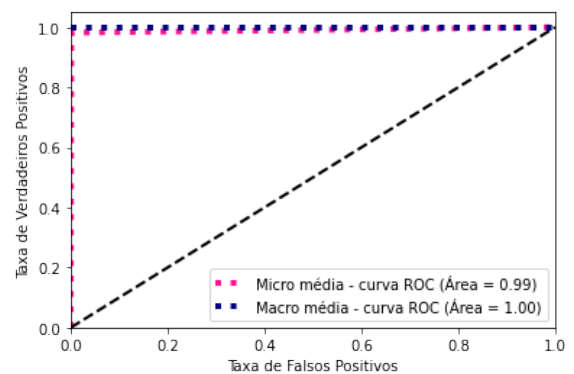
Configuração das cargas	Precisão	Revocação	F1	Quantidade de amostras
0	1,00	1,00	1,00	122
1	1,00	1,00	1,00	125

2	1,00	1,00	1,00	129
3	1,00	1,00	1,00	115
4	0,93	0,99	0,96	108
5	0,91	0,89	0,90	122
6	0,88	0,93	0,90	121
7	1,00	0,93	0,96	137
8	1,00	0,98	0,99	112
9	1,00	1,00	1,00	114
10	1,00	1,00	1,00	124
11	1,00	1,00	1,00	114
12	1,00	1,00	1,00	115
13	1,00	1,00	1,00	127
14	1,00	1,00	1,00	108
15	1,00	1,00	1,00	127

Fonte: Produção do(a) próprio(a) autor(a).



(a)



(b)

Figura 6: (a) Matriz de confusão do experimento 4. (b) Curva ROC do experimento 4.

Tabela 7 – Experimento 5: Resultados dos critérios (precisão, revocação e métrica F1) de teste das *embeddings* da base de cargas distintas 2 (Base D2, Tabela 2) do 1D-CAE com o KNN, sem o recurso de *shifting*.

Configuração das cargas	Precisão	Revocação	F1	Quantidade de amostras
0	1,00	1,00	1,00	122
1	0,88	0,75	0,81	125
2	0,80	0,78	0,79	129
3	0,99	1,00	0,99	115
4	0,93	1,00	0,96	108
5	1,00	1,00	1,00	122
6	0,97	0,99	0,98	121
7	0,83	0,80	0,81	137
8	0,51	0,56	0,53	112
9	1,00	1,00	1,00	114
10	0,70	0,62	0,66	124
11	0,81	0,91	0,86	114
12	0,75	0,72	0,74	115
13	1,00	1,00	1,00	127
14	1,00	0,99	0,99	108
15	0,93	1,00	0,97	127

Fonte: Produção do(a) próprio(a) autor(a).

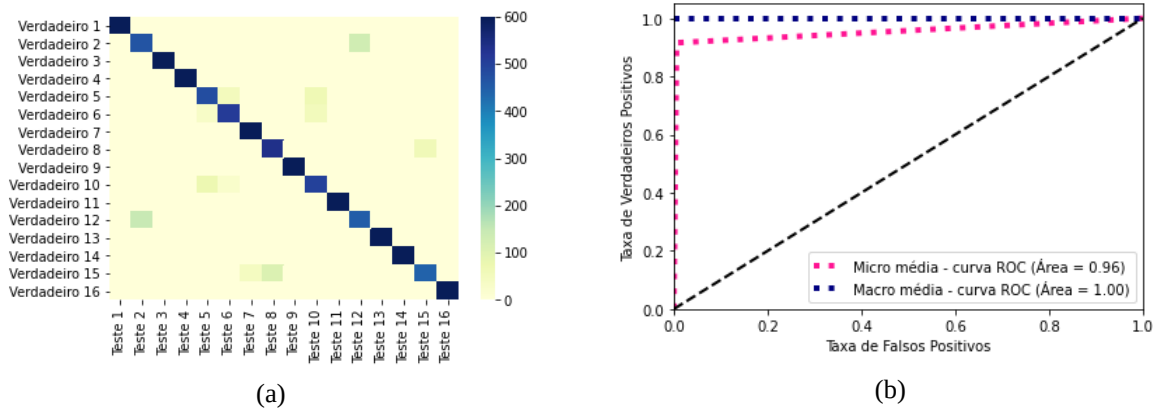


Figura 7: (a) Matriz de confusão do experimento 5. (b) Curva ROC do experimento 5.

Por último, os dados foram armazenados no sistema gerenciador de bancos de dados (SGBD) MongoDB, um banco de dados orientado a documentos. O banco de dados guarda 5 informações por documento: o nome da base de cargas (e.g. Base D1, Base S1), a numeração da configuração das cargas (0 até 15), os valores das *embeddings* (um vetor de 170 números), a data de armazenamento e o autor da coleta. O experimento foi realizado com os dados da Base D1, o armazenamento de 9600 documentos foi realizado em 7,21 segundos e o banco de dados de *embeddings* ocupa apenas 20,4MB de memória. Para comparação, o repositório contendo os dados originais das cargas em formato CSV possui 834MB, ou seja, cerca de 40,88 vezes mais unidades de memória ocupados.

6 Conclusões

Por meio do trabalho realizado, a primeira observação feita é sobre o potencial das redes neurais em problemas envolvendo dados não estruturados e com configurações complexas. Esses algoritmos desempenham muito acima dos outros algoritmos clássicos de ML conforme a complexidade e a quantidade de dados aumenta (AGGARWAL et al., 2018). Um fator determinante para o sucesso na extração de características foi a quantidade de dados que foram utilizados no treinamento do modelo. Assim, por meio dos resultados obtidos, o algoritmo de 1D-CAE apresentou elevada eficiência e acurácia na extração de características que podem ser consumidas por modelos mais simples e de rápido treinamento (no caso do projeto, utilizando apenas 10 amostras de características de cargas no treinamento).

A similaridade de cossenos não indicou ser uma forma viável de classificar as cargas, embora tenha apresentado melhoras de acurácia após a aplicação de *shifting*. Contudo, em trabalhos futuros, podem ser testadas outras métricas de similaridade.

O armazenamento das características pode ser feito tanto para arquivos como para um banco de dados. Além das características serem eficientes para o treinamento de modelos e apresentarem resultados com elevada acurácia, elas ainda podem ser salvas mais rapidamente e em maior quantidade conforme a massa de dados cresce, visto que representam aproximadamente 10,2% do tamanho dos dados de carga originais. Ressaltando que, conforme os experimentos, os dados de *embeddings* armazenados no MongoDB ocupam cerca de 2,45% da quantidade de memória que os dados originais ocupam em formato CSV.

Um dos objetivos para o projeto não foi concluído em decorrência da complexidade do projeto para o tempo planejado. A identificação de novas cargas inseridas na rede elétrica representa um desafio. Observa-se

ser um caso em que é necessária uma forma de aprendizado baseada em similaridade com os dados do banco de dados, uma tarefa que não demanda apenas muitos dados, mas uma enorme variedade de dados. A implementação supervisionada de um algoritmo para tal iria contrário à um dos objetivos do projeto: reduzir custos de produção de modelos de IA complexos por residência. Esse objetivo foi atingido na propostas cobertas pelo projeto, com apenas um modelo que gera características em baixa dimensionalidade e a criação de um banco de dados que armazena essas características, possibilitando a utilização de modelos simples por residência, que aprendem com as características dos dados, e permitindo um armazenamento mais leve e escalável dos dados.

Referências Bibliográficas

- [1] BATRA, Nipun et al. Data driven energy efficiency in buildings. **arXiv preprint arXiv:1404.7227**, 2014.
- [2] KIM, Jihyun; LE, Thi-Thu-Huong; KIM, Howon. Nonintrusive load monitoring based on advanced deep learning and novel signature. **Computational intelligence and neuroscience**, v. 2017, 2017.
- [3] CHEN, Shumei; YU, Jianbo; WANG, Shijin. One-dimensional convolutional auto-encoder-based feature learning for fault diagnosis of multivariate processes. **Journal of Process Control**, v. 87, p. 54-67, 2020.
- [4] ALI, Syed Saqib; CHOI, Bong Jun. State-of-the-art artificial intelligence techniques for distributed smart grids: A review. **Electronics**, v. 9, n. 6, p. 1030, 2020.
- [5] ZHANG, Dongxia; HAN, Xiaoqing; DENG, Chunyu. Review on the research and practice of deep learning and reinforcement learning in smart grids. **CSEE Journal of Power and Energy Systems**, v. 4, n. 3, p. 362-370, 2018.
- [6] MONACCHI, Andrea et al. GREEND: An energy consumption dataset of households in Italy and Austria. In: **2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)**. IEEE, 2014. p. 511-516.
- [7] MAKONIN, Stephen et al. AMPds: A public dataset for load disaggregation and eco-feedback research. In: **2013 IEEE electrical power & energy conference**. IEEE, 2013. p. 1-6.
- [8] KOLTER, J. Zico; JOHNSON, Matthew J. REDD: A public data set for energy disaggregation research. In: **Workshop on data mining applications in sustainability (SIGKDD)**, San Diego, CA. 2011. p. 59-62.
- [9] BARKER, Sean et al. Smart*: An open data set and tools for enabling research in sustainable homes. **Sust KDD**, August, v. 111, n. 112, p. 108, 2012.
- [10] BELLMAN, RICHARD. Dynamic programming, princeton univ. **Press Princeton, New Jersey**, 1957.
- [11] RUSSELL, Stuart; NORVIG, Peter. Artificial intelligence: a modern approach. 2002.
- [12] AGGARWAL, Charu C. et al. Neural networks and deep learning. **Springer**, v. 10, p. 978-3, 2018.
- [13] CHEN, Shumei; YU, Jianbo; WANG, Shijin. One-dimensional convolutional auto-encoder-based feature learning for fault diagnosis of multivariate processes. **Journal of Process Control**, v. 87, p. 54-67, 2020.
- [14] KRAMER, Mark A. Nonlinear principal component analysis using autoassociative neural networks. **AICHE journal**, v. 37, n. 2, p. 233-243, 1991.
- [15] RUMELHART, David E.; HINTON, Geoffrey E.; WILLIAMS, Ronald J. Learning representations by back-propagating errors. **nature**, v. 323, n. 6088, p. 533-536, 1986.
- [16] SOARES, Wallyson Carvalho et al. Deep Learning Aplicado à Identificação de Cargas Elétricas Similares em Smart Grid. **Latin American Journal of Energy Research**, v. 6, n. 1, p. 1-11, 2019.
- [17] FIRMES, Victor Pereira et al. Identificação Nao-intrusiva de Cargas Similares em Smart Grid Usando Rede Neural Convolutacional. **Anais do 14º Simpósio Brasileiro de Automação Inteligente (SBAI)**, 2019.
- [18] PAIXAO, A. et al. Classificação Inteligente Aplicada ao Problema de Identificação de Cargas Elétricas “Idênticas”. **Simpósio Brasileiro de Pesquisa Operacional, Anais do XLVIII SBPO**, 2016.
- [19] ANOWAR, Farzana; SADAoui, Samira; SELIM, Bassant. Conceptual and empirical comparison of dimensionality reduction algorithms (PCA, KPCA, LDA, MDS, SVD, LLE, ISOMAP, LE, ICA, t-SNE). **Computer Science Review**, v. 40, p. 100378, 2021.
- [20] GOODFELLOW, Ian et al. Generative adversarial nets. **Advances in neural information processing systems**, v. 27, 2014.
- [21] ROKACH, Lior. Ensemble-based classifiers. **Artificial intelligence review**, v. 33, n. 1, p. 1-39, 2010.
- [22] WOLPERT, David H. Stacked generalization. **Neural networks**, v. 5, n. 2, p. 241-259, 1992.
- [23] GÉRON, Aurélien. **Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems**. O'Reilly Media, 2019.
- [24] MURPHY, Kevin P. **Machine learning: a probabilistic perspective**. MIT press, 2012.

- [25] FIX, Evelyn; HODGES, Joseph Lawson. Discriminatory analysis. Nonparametric discrimination: Consistency properties. **International Statistical Review/Revue Internationale de Statistique**, v. 57, n. 3, p. 238-247, 1989.
- [26] KOHAVI, Ron et al. Bias plus variance decomposition for zero-one loss functions. In: **ICML**. 1996. p. 275-83.
- [27] MONGODB. **MongoDB Community Server**. Disponível em: <https://www.mongodb.com/try/download/community>. Acesso em: 14 set. 2021.
- [28] FERRI, César; HERNÁNDEZ-ORALLO, José; SALIDO, Miguel Angel. Volume under the ROC surface for multi-class problems. In: **European conference on machine learning**. Springer, Berlin, Heidelberg, 2003. p. 108-120.
- [29] LECUN, Yann et al. Backpropagation applied to handwritten zip code recognition. **Neural computation**, v. 1, n. 4, p. 541-551, 1989.