



UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Rafael Barbosa de Oliveira

**Arquitetura de Modelos Híbridos, Machine
Learning e Otimizadores para Análise de
Consumo de Energia Elétrica e Produtividade
em Pintura Automotiva**

Belém

2024


**“PREDIÇÃO DE CONSUMO DE ENERGIA ELÉTRICA E PRODUTIVIDADE EM
OFINCA DE PINTURA AUTOMOTIVA”**

AUTOR: RAFAEL BARBOSA DE OLIVEIRA


DISSERTAÇÃO DE MESTRADO SUBMETIDA À BANCA EXAMINADORA APROVADA PELO
COLEGIADO DO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA, SENDO
JULGADA ADEQUADA PARA A OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA
ELÉTRICA NA ÁREA DE COMPUTAÇÃO APLICADA.

APROVADA EM: 27/03/2024


BANCA EXAMINADORA:

Documento assinado digitalmente
 **Roberto Celio Limao de Oliveira**
Data: 27/03/2024 12:40:47-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Roberto Célio Limão de Oliveira
(Orientador – PPGEE/ITEC/UFPA)

Documento assinado digitalmente
 **MARCOS CESAR DA ROCHA SERUFFO**
Data: 27/03/2024 13:12:01-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Marcos César da Rocha Seruffo
(Avaliador Interno – PPGEE/ITEC/UFPA)

Documento assinado digitalmente
 **JEFFERSON MAGALHAES DE MORAIS**
Data: 27/03/2024 12:58:22-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Jefferson Magalhães de Moraes
(Avaliador Externo ao Programa – PPGCC/UFPA)

VISTO:

Prof. Dr. Diego Lisboa Cardoso
(Coordenador do PPGEE/ITEC/UFPA)

**Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD
Sistema de Bibliotecas da Universidade Federal do Pará
Gerada automaticamente pelo módulo Ficat, mediante os dados fornecidos pelo(a) autor(a)**

B238a Barbosa de Oliveira, Rafael.
Arquitetura de Modelos Híbridos, Machine Learning e
Otimizadores para Análise de Consumo de Energia Elétrica e
Produtividade em Pintura Automotiva / Rafael Barbosa de Oliveira.
— 2024.
70 f. : il. color.

Orientador(a): Prof. Dr. Roberto Célio Limão de Oliveira
Dissertação (Mestrado) - Universidade Federal do Pará,
Instituto de Tecnologia, Programa de Pós-Graduação em
Engenharia Elétrica, Belém, 2024.

1. Energia. 2. Pintura Automotiva. 3. Aprendizado de
Máquina. 4. Meta-heurística. 5. Modelo Híbrido. I. Título.

CDD 006.3

Aos meus Pais e irmãos, minhas estruturas para os desafios na vida.

Agradecimentos

Gratidão à Deus que sempre esteve em tudo e em cada detalhe da minha história.

À toda minha família e à Christiane Aleixo, que estiveram comigo com contribuições indispensáveis.

Ao meu orientador, Professor Roberto Célio Limão de Oliveira a quem sou muito grato pelas orientações, confiança e gigantes colaborações em minha vida pessoal e profissional que vão além das conquistas nesta pós-graduação. Mais uma vez minha eterna gratidão. Ao meu mentor que sempre esteve comigo nos diversos desafios enfrentados, Professor Helder Aranha.

Aos meus amigos Elielma, Riete e Lenilce pela torcida e ombro amigo. E um agradecimento especial à Ezequiel Assis pela grande força.

À Softex, aqui representada pelas pessoas Barbara Marciel, Vic Ribeiro e David Moyses, ambiente e pessoas de auto nível de colaboração e aprendizado e que contribuíram muito para esse momento.

Aos meus nobres colegas da Residência em Dados e Analytics da Universidade de Pernambuco e da Stellants/Jeep de Goiana/PE que contribuíram com todo o processo de diversas formas, ao Paulo Couto, Victor Kirschner e Thiago Cavalcanti, bem como a pessoa do coordenador Alexandre Maciel.

A todos que contribuíram, direta ou indiretamente, para o desenvolvimento deste trabalho.

*"Procure essas coisas, como se procurasse prata ou um tesouro escondido. Se você fizer isso, saberá o que quer dizer temer o SENHOR, e aprenderá a conhecê-lo. É o SENHOR quem dá sabedoria; a sabedoria e o entendimento vêm dele."
(Bíblia, Provérbios 2:4-6)*

Resumo

Estratégias de otimização de consumo energético nas etapas de pintura emergem como fatores primordiais para promover uma produção mais sustentável e competitiva no setor automotivo. Esta Dissertação busca prever o consumo energético e maximizar a produtividade na pintura automotiva, utilizando uma abordagem que combina seleção de variáveis, modelos híbridos, hiperparâmetros destes modelos e otimização por meta-heurística em uma arquitetura com 3 etapas. Os processos de pintura automotiva apresentam variáveis em forma de séries temporais que descrevem o histórico do consumo de energia. Na etapa 1, escolhe-se o melhor modelo de aprendizado de máquina (Random Forest, Long-Short Term Memory, XGBoost e GRU-LSTM) para prever séries temporais do consumo energético em $t+1$. Na etapa 2, avalia-se os modelos RF, XGBoost e Rede Neural Artificial (RNA) Densa para selecionar o melhor preditor de quantidade de veículos produzidos (ciclos). Na etapa 3, seleciona-se a melhor meta-heurística entre Genetic Algorithm (GA), Differential Evolution (DE) e Particle Swarm Optimization (PSO) para otimizar o consumo energético previsto pelo melhor modelo do step 1, usando como medida de *fitness* o melhor modelo do step 2. A arquitetura final reduziu a energia consumida em até 16% e aumentou o ciclo em 127%, usando os modelos GRU-LSTM na etapa 1, RNA Densa na etapa 2 e DE no etapa 3. Os resultados ressaltam a oportunidade de uso da abordagem proposta para otimizar o consumo de energia e a produtividade na pintura automotiva.

Palavras-chave: Energia. Manufatura. Pintura Automotiva. Aprendizado de Máquina. Meta-heurística. Modelo Híbrido.

Abstract

Strategies for optimizing energy consumption in the painting stages are emerging as key factors in promoting more sustainable and competitive production in the automotive sector. This dissertation seeks to predict energy consumption and maximize productivity in automotive painting, using an approach that combines variable selection, hybrid models, hyperparameters of these models and meta-heuristic optimization in a 3-stage architecture. Automotive painting processes have variables in the form of time series that describe the history of energy consumption. In stage 1, the best machine learning model is chosen (Random Forest, Long-Short Term Memory, XGBoost and GRU-LSTM) to predict energy consumption time series at $t+1$. In step 2, the RF, XGBoost and Dense Artificial Neural Network (ANN) models are evaluated to select the best predictor of the number of vehicles produced (cycles). In step 3, the best metaheuristic between Genetic Algorithm (GA), Differential Evolution (DE) and Particle Swarm Optimization (PSO) is selected to optimize the energy consumption predicted by the best model from step 1, using the best model from step 2 as a fitness measure. The final architecture reduced the energy consumed by up to 16% and increased the cycle by 127%, using the GRU-LSTM models in step 1, Dense ANN in step 2 and DE in step 3. The results highlight the opportunity to use the proposed approach to optimize energy consumption and productivity in automotive painting.

Keywords: Energy. Manufacturing. Automotive Painting. Machine Learning. Meta-heuristics. Hybrid Model.

Lista de ilustrações

Figura 1.	Linha de produção automotiva	16
Figura 2.	Neurônio Artificial Perceptron.	27
Figura 3.	Curvas das funções σ , ReLU, \tanh e <i>LeakyReLU</i>	28
Figura 4.	Exemplo de estrutura de uma rede neural densa de 5 camadas.	29
Figura 5.	Arquitetura da LSTM.	32
Figura 6.	Arquitetura da GRU.	34
Figura 7.	Arquitetura do modelo híbrido.	34
Figura 8.	Pseudocódigo do Modelo Híbrido GRU-LSTM	35
Figura 9.	Pseudocódigo do GA	36
Figura 10.	Pseudocódigo do PSO	37
Figura 11.	Pseudocódigo do DE	39
Figura 12.	Arquitetura Final	43
Figura 13.	Fluxograma ETL, Análise Exploratória e Pré-processamento de dados.	44
Figura 14.	Fluxograma - Etapa ETL	46
Figura 15.	Violinplot Consumo (Kwh) & Ciclos.	47
Figura 16.	Consumo diário de Kwh de janeiro a setembro de 2023.	47
Figura 17.	Dados null para a variável ciclos.	48
Figura 18.	Violinplot consumo (Kwh) por processos de produção.	49
Figura 19.	Correlação de Pearson.	50
Figura 20.	<i>Step 1</i> da arquitetura proposta.	51
Figura 21.	Variação do erro em função do número de <i>steps</i>	51
Figura 22.	<i>Step 2</i> da arquitetura proposta.	52
Figura 23.	<i>Step 3</i> da arquitetura proposta.	53
Figura 24.	Ajuste das variáveis para o modelo híbrido.	55
Figura 25.	Análise residual da GRU-LSTM	56
Figura 26.	Predição do ciclo de produção utilizando o modelo DL.	57
Figura 27.	Curvas de convergência: modelo RNA Densa.	58
Figura 28.	Análise residual da RNA Densa	59
Figura 29.	Arquitetura Final	62

Lista de quadros

Quadro 1.	Dicionário e descrição dos dados	44
Quadro 2.	Especificações de <i>hardware</i> e <i>software</i> utilizados no projeto.	45

Lista de tabelas

Tabela 1.	Hiperparâmetros dos modelos e otimizadores.	52
Tabela 2.	Avaliação dos modelos no <i>step 1</i>	54
Tabela 3.	Avaliação dos modelos no <i>step 2</i>	57
Tabela 4.	Desempenho dos otimizadores no <i>step 3</i>	60
Tabela 5.	Combinação de todos os modelos do <i>step 2</i> e <i>step 3</i>	60

Lista de abreviaturas e siglas

AutoML	<i>Automated Machine Learning</i>
CNN	<i>Convolutional Neural Network</i>
DE	<i>Differential Evolution</i>
DL	<i>Deep Learning</i>
EDA	<i>Exploratory Data Analysis</i>
EE	Energia Elétrica
EEMD	<i>Ensemble Empirical Mode Decomposition</i>
EVS	<i>Explained Variance Score</i>
ETL	<i>Extract, Transform and Load</i>
GA	<i>Genetic Algorithm</i>
GRU	<i>Gated Recurrent Unit</i>
IA	Inteligência Artificial
IoT	<i>Internet of Things</i>
LeakyReLU	<i>Inclined Rectified Linear Unit</i>
LSTM	<i>Long Short-Term Memory</i>
MAE	<i>Mean Absolute Error</i>
MAPE	<i>Mean Absolute Percentual Error</i>
MSE	<i>Mean Square Error</i>
ML	<i>Machine Learning</i>
NLP	<i>Natural Language Processing</i>
ONU	Organização das Nações Unidas
PSO	<i>Particle Swarm Optimization</i>
RF	<i>Random Forest</i>

RNA	Rede Neural Artificial
RNN	<i>Recurrent Neural Network</i>
ReLU	<i>Rectified Linear Unit</i>
SQL	<i>Structured Query Language</i>
XGBoost	<i>Extreme Gradient Boosting</i>

Sumário

1	INTRODUÇÃO	15
1.1	Descrição do Problema	15
1.2	Trabalhos Relacionados	17
1.3	Justificativa	19
1.4	Objetivo	20
1.5	Objetivos Específicos	20
1.6	Organização da Dissertação	21
2	FUNDAMENTAÇÃO TEÓRICA	22
2.1	Área do Negócio	22
2.2	Modelos de Inteligência Computacional	23
2.2.1	<i>Random Forest</i>	24
2.2.2	<i>Extreme Gradient Boosting</i>	24
2.2.3	Redes Neurais Artificiais	26
2.2.4	<i>Long Short Term Memory</i>	31
2.2.5	<i>Gated Recurrent Unit</i>	33
2.2.6	Modelo Híbrido GRU-LSTM	33
2.2.7	<i>Genetic Algorithm</i>	35
2.2.8	<i>Particle Swarm Optimization</i>	37
2.2.9	<i>Differential Evolution</i>	38
2.3	Avaliação de Desempenho dos de ML	40
3	MATERIAL E MÉTODOS	42
3.1	Metodologia Proposta	42
3.1.1	Step 1 - Predição do consumo energético	42
3.1.2	Step 2 - Predição do ciclo de produção	42
3.1.3	Step 3 - Otimização utilizando algoritmos evolutivos	42
3.2	Banco de Dados: ETL, Análise Exploratória e Pré-processamento de dados	44
3.2.1	Extração, Transformação e Carga	46
3.2.2	Período dos Dados Coletados	46
3.2.3	Tratamento de Valores Ausentes	47
3.2.4	<i>Os outliers</i>	48
3.2.5	Seleção de variáveis	48
3.3	Desenvolvimento da Arquitetura Proposta	49
3.3.1	Step 1 - Predição do consumo energético	49

3.3.2	<i>Step 2 - Predição Do Ciclo De Produção</i>	52
3.3.3	<i>Step 3 - Otimização Utilizando Algoritmos Evolutivos</i>	52
4	RESULTADOS E DISCUSSÃO	54
4.1	Resultados do <i>Step 1</i>	54
4.2	Resultados do <i>Step 2</i>	56
4.3	Resultados do <i>Step 3</i>	58
4.4	Arquitetura Final e discussões	61
5	CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS	64
	REFERÊNCIAS	66

1 Introdução

Tecnologias que sejam capazes de prever consumos de energia estão assumindo destaque e relevância na academia, mercado e sociedade. Estas tecnologias quando aplicadas à manufatura podem reduzir custos operacionais e trazer eficiência no processo de produção, gerando redução de consumo das potências geradas a partir de gás, eletricidade e ar comprimido, principais recursos energéticos consumidos nas grandes indústrias de produção de veículos. Na indústria de automotivos os desafios são diversos. Com processos automatizados para atender uma produção contínua, ter sistemas capazes de prever o consumo energético tem grande relevância para o setor, contribuindo para uma efetividade maior nos ciclos produtivos (produtividade de veículos) em todas as etapas do processo .

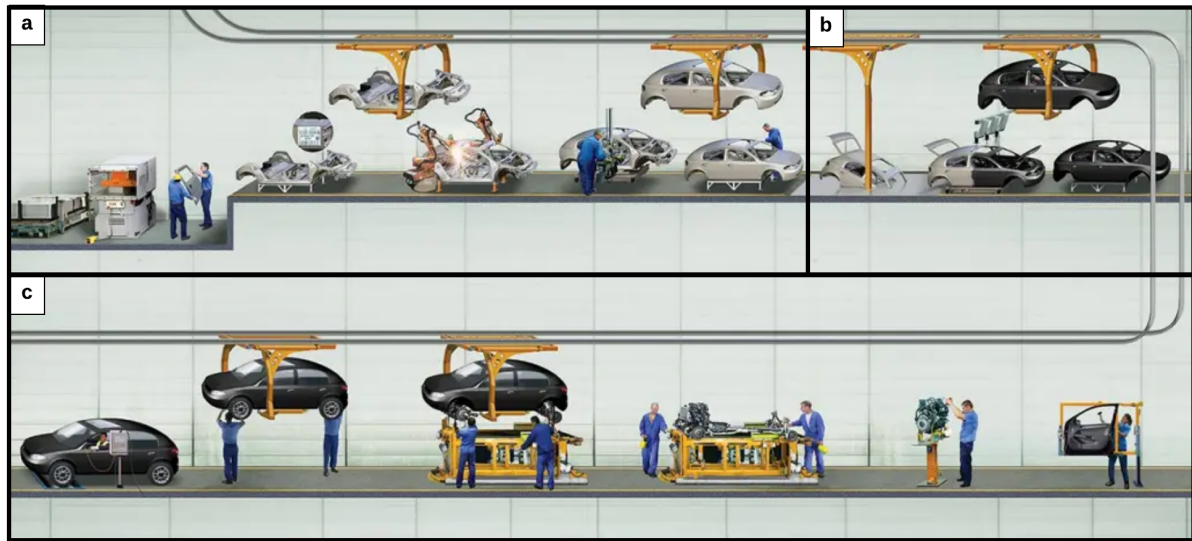
Um sistema de predição capaz de antecipar o total de energia consumida nos próximos minutos, horas ou dias nos diferentes processos de produção (nível macro) ou ainda, antecipar o total de consumo de cada máquina em cada processo (nível micro) resultaria em benefícios significativos para times operacionais e de níveis estratégicos nas organizações, podendo impactar diretamente na redução de custos e programação de operações que exigem mais energia nas oficinas de pintura, a exemplo das câmaras de secagem conforme afirma Ma, Jiang e Liu (2019). Para o autor, tais sistemas podem permitir ajustes nos processos de produção e gerenciamento de recursos de forma mais eficaz das empresas.

A indústria automotiva possui ao menos três etapas na produção de veículos que são estruturação ou funilaria (Figura 1.a), pintura (Figura 1.b) e montagem (Figura 1.c). Destas 3 etapas, a pintura é a maior consumidora de energia e seus processos podem consumir em torno de 50% do consumo total de energia e o custo pode chegar até a 10% da produção de um veículo (OKTAVIANDRI; SAFIEE, 2018; YANG et al., 2021).

1.1 Descrição do Problema

A ineficiência no consumo de energia na oficina de pintura impacta diretamente no total de energia consumida, no custo direto do consumo da planta industrial excessivo e aumentando os custos de produção e, também aumentando ainda mais os impactos ambientais. Com processos de automatização constante, esta manufatura tem investido bastante recursos em tecnologias que executam ações cada vez mais sem intervenções humanas e conseqüentemente com uma tendência de ter mais consumo energético além do já consumido hoje, tendo uma participação ainda maior em grande parte do consumo de energia no mundo. O alto nível e o avanço das automatizações têm intensificado a complexidade dos processos de cada etapa da produção automotiva, pois estes possuem

Figura 1. Linha de produção automotiva



Fonte: Adaptado de Carvalho (2017)

arranjos de diversas máquinas, as quais são as responsáveis pelo o consumo de energia na planta industrial.

Além dos fatores diretamente relacionados à planta industrial que impactam no consumo de energia, existem outros fatores que podem impactar a eficiência energética. A oficina de pintura (local da etapa de pintura) pode sofrer influência da temperatura e umidade do ambiente, aumentando ou reduzindo o tempo de secagem e cura da tinta. A pressão excessiva ou baixa do ar pode gerar desperdício de tinta e qualidade inferior no acabamento dos veículos produzidos (GIAMPIERI et al., 2020).

A oficina de pintura possui processos que contém máquinas especializadas em ações automatizadas e complexas com desafios a serem superados quando tem-se como estratégia a busca por uma gestão estratégica com base em previsão do consumo de energia. Um dos principais insumos para estas previsões são bancos de dados com registros de consumos de cada máquina em um fluxo contínuo de coleta e armazenamento dos dados, porém, atualmente, as plantas de produção não possuem uma estrutura de dados que atenda esta necessidade, sendo um dos fatores inibidores à implementação de estratégias eficazes de previsão nas oficinas de pintura.

Com o crescente interesse no mercado em inteligência artificial, com foco em Aprendizado de Máquina (do inglês *Machine Learning* - ML) e Aprendizado Profundo (do inglês *Deep Learning* - DL), tem aumentado as demandas por modelos de predição em diversas áreas, com destaque para a eficiência energética, o que tem sido um fator favorável para o investimento em infraestrutura e arquitetura de dados, favorecendo a superação dos obstáculos para a construção de modelos preditivos para aprimorar a eficiência energética

e operacional na indústria automotiva. (ĐAKOVIĆ et al., 2023).

Estudos têm aplicado diversas técnicas técnicas de ML e DL na manufatura para eficiência energética, como a Floresta Aleatória (do inglês Random Forest - RF) (WANG et al., 2018) e XGBoost (FATAHI, 2022) na para prever o consumo de energia na indústria. Para a área de DL, tem-se de diversas publicações com Redes Neurais Artificiais Densas e uma exploração de modelos ainda mais robustos como as redes neurais recorrentes, como as redes Memória de longo e curto prazo (do inglês *Long Short-Term Memory* - LSTM). Sua utilização tem sido motivada por sua capacidade de capturar padrões que possuem maior grau de complexidade em problemas com dados tabulares, de aprendizado de linguagens, como o Processamento de Linguagem Natural (do inglês *Natural Language Processing* - NLP), e ainda tem apresentado níveis altos de efetividade trabalhando com dados de consumo de energia (JIANG et al., 2023).

Além dos modelos de ML e DL, outras abordagens estão sendo experimentadas, buscando extrair ainda mais eficácia dos modelos, tais abordagens são chamadas de modelos híbridos, que juntam os melhores resultados de dois ou mais modelos de ML e DL distintos e um modelo único. Os resultados dos modelos híbridos mostram-se promissores em análises baseadas, na linha na problemática deste trabalho, problema de predição com dados de séries temporais (BASHIR et al., 2022; ZAFAR et al., 2022). Autores, apontam que estes modelos híbridos podem promover previsões mais precisas e *insights* mais profundos sobre o comportamento do consumo de energia em oficinas de pintura (LU et al., 2023).

Um horizonte promissor está sendo construído ante aos diversos desafios e possibilidades que estão sendo impulsionados por avanços tecnológicos e uma crescente conscientização ambiental. À medida que a indústria automotiva se adapta ao discurso sustentável, a pintura de veículos emerge como uma etapa crucial não apenas na produção, mas também como um campo de oportunidades para inovações que integram eficiência operacional e responsabilidade ambiental. O futuro da pintura automotiva se configura como um capítulo no qual a sinergia entre qualidade, eficiência e sustentabilidade determinará o rumo da indústria, impulsionando-a em direção a um modelo mais equilibrado e alinhado com as exigências contemporâneas.

1.2 Trabalhos Relacionados

Na academia tem-se utilizado técnicas para extrair o potencial máximo de técnicas como mineração de dados, ML e DL para predição do consumo energético, melhorando a definição de estratégias de produção e reduzindo os custos operacionais em oficinas de pintura automotiva. O trabalho de Cavalcante et al. (2020) apresenta uma metodologia que ajusta um modelo matemático para prever o comportamento do processo de cura de tintas. Com a metodologia é possível determinar os coeficientes de transferência de

calor. Os resultados indicam que o modelo é uma ferramenta promissora para desenvolver estratégias de otimização operacional, além de minimizar retrabalhos e custos de energia. Todo o modelo foi construído tendo como base, os dados coletados na indústria.

Estudos com modelos híbridos têm surgido para trazer maior eficácia aos modelos de ML. O estudo de Lu et al. (2023) traz uma abordagem baseada em ML híbrida definida como Prophet-EEMD-LSTM. O primeiro modelo implementado é o Prophet que tem como principal finalidade ajustar as características cíclicas e de tendência dos dados. A seguir, o código é estruturado para receber o *Ensemble Empirical Mode Decomposition* (EEMD) que atua decompondo os resíduos e melhorando a sensibilidade do modelo. A LSTM utiliza os resíduos da etapa anterior para capturar as características temporais dos dados. Os resultados experimentais evidenciaram o alto desempenho preditivo do modelo híbrido Prophet-EEMD-LSTM, com valor médio de R^2 0,94006.

O trabalho publicado por Hovgard, Lennartson e Bengtsson (2022) investiga a redução do uso de energia com foco direcionado às estações robóticas em linhas de produção automotiva. O autor aplicou técnicas de otimização estocástica e obteve como resultado uma redução de até 23% do consumo de energia garantindo uma grande efetividade energética e variação eficaz do tempo de ciclo de produção.

Para além da indústria automotiva, Ilbeigi, Ghomeishi e Dehghanbanadaki (2020) desenvolveram pesquisas de otimização do consumo de energia para o consumo de eletricidade em edifícios. Os autores propuseram um método de ML que utiliza uma abordagem que combina modelagem com o software EnergyPlus, Rede Neural Artificial (RNA) e otimização com Algoritmo Genético (do inglês *Genetic Algorithm* - GA). Os resultados apresentados no trabalho demonstram uma redução de cerca de 35% no consumo de energia após a implementação de otimização do sistema.

Narciso e Martins (2020) realizaram um levantamento da literatura analisando 42 artigos voltados a estudos sobre eficiência energética na indústria. Seus levantamentos mostram que no campo da previsão de consumo energético, a maioria das contribuições visam obter modelos preditivos para o consumo de energia, com ênfase na correspondência entre oferta e demanda de energia. Por outro lado, na otimização do consumo de energia, a maioria dos estudos se concentram na resolução de problemas de otimização, buscando insumos ideais para melhorar a eficiência energética.

As abordagens apresentadas e outras semelhantes expandem o conhecimento científico em produções que produzem ampla variedade de métodos inovadores para analisar, otimizar e prever o consumo de energia, possibilitando melhorias substanciais na eficiência energética e na sustentabilidade nos diversos setores como a indústria automotiva, edificações residenciais e comerciais, entre outros. Essas práticas não só impulsionam avanços tecnológicos, mas também contribuem para a preservação do meio ambiente e para a promoção de um futuro mais sustentável.

O diferencial desta pesquisa em relação aos trabalhos apresentados reside na aplicação de uma arquitetura de 3 steps genérica, integrando técnicas de ML, DL e algoritmos meta-heurísticos. Apesar de ser experimentada e validada no setor de pintura automotiva, essa arquitetura pode ser aplicada em diversos outros setores, representando assim uma contribuição significativa para diferentes campos de estudo.

1.3 Justificativa

A utilização de inovação tecnológica na etapa de manufatura de veículos está ajudando a impulsionar a indústria automotiva rumo a um futuro mais econômico e sustentável, pois atua na redução de recursos operacionais, ampliação de rendimentos e redução de consumo de recursos naturais.

A adoção de técnicas metodológicas robustas no processo de pintura automotiva reduz o dispêndio de energia sem comprometer a eficiência do produto final. Através do *benchmarking* de projetos e soluções em eficiência energética na pintura automotiva, é possível analisar casos de sucesso de empresas que já implementaram medidas nesse sentido e seus resultados em larga escala. Essa técnica de gestão proporciona a adoção das melhores práticas de mercado e a busca constantemente a excelência, no qual as empresas podem aumentar sua competitividade e alcançar o sucesso a longo prazo.

A sustentabilidade na indústria automotiva é um tema cada vez mais relevante, considerando os desafios ambientais e a necessidade de reduzir o impacto negativo dessa indústria no meio ambiente. Diversas medidas e práticas estão sendo adotadas para tornar a indústria automotiva mais sustentável, abrangendo desde as etapas de produção dos veículos até o seu uso e descarte. Szász, Csíki e Rácz (2021), destaca a importância de uma perspectiva minuciosa para compreender as condutas das empresas automotivas que visam a sustentabilidade. O estudo ressalta que, para entender completamente como as empresas do setor automotivo estão lidando com questões ambientais, sociais e econômicas, é preciso considerar diversos fatores, como a complexidade da cadeia de suprimentos, a influência de diferentes *stakeholders* e as estratégias de gestão adotadas pelas empresas.

Ademais, Lukin, Krajnović e Bosna (2022) mostra como as relevantes empresas do setor automotivo estão definindo seus processos e operações para alcançar os objetivos de desenvolvimento sustentável definidos pela Organização das Nações Unidas (ONU) para 2030, destacando que as empresas estão cada vez mais reconhecendo a importância da sustentabilidade e estão integrando princípios sustentáveis em suas práticas de negócios.

As empresas do setor automotivo estão cientes que a sustentabilidade não é apenas uma questão de responsabilidade social, mas também uma oportunidade de negócio, uma vez que os consumidores estão cada vez mais valorizando empresas que adotam práticas sustentáveis, faz-se necessário a realização de um estudo visando, além da previsão do

consumo energético, a potencialização do ciclo de produção a partir dos prognósticos obtidos, de modo obter gestão eficiente de energia, a sustentabilidade e a eficiência operacional na indústria automotiva.

1.4 Objetivo

Este trabalho tem como objetivo prever o consumo energético e maximizar o ciclo de produção em um ambiente de pintura automotiva a partir de uma arquitetura de 3 *steps*, utilizando técnicas de ML, DL e algoritmos meta-heurísticos.

1.5 Objetivos Específicos

Os objetivos específicos estão alinhados com o objetivo geral desta pesquisa e desempenham um papel relevante ao contribuir para a concretização deste último, potencializando assim os resultados almejados.

- Levantar base de dados de consumo dos vetores e ciclos de produção;
- Extrair, Transformar e Carregar (do inglês *Extract, Transform and Load* - ETL) os dados em uma base de dados;
- Realizar Análise Exploratória de Dados (do inglês *Exploratory Data Analysis* - EDA);
- Avaliar um possível impacto no consumo para cada modelo produzido;
- Desenvolver modelos de predição com base nas diferentes grandezas que impactam no consumo;
- Construir arquitetura de três *steps* com os melhores modelos de predição e otimização para a Unidade Operativa de Pintura;
- Desenvolver modelos para o *step* 1 para prever o Consumo Energético em $t+1$ com base em dados de séries temporários;
- Desenvolver modelos de regressão para o *step* 2 para prever ciclos de produção a partir do consumo energético;
- Desenvolver modelos meta-heurísticos de maximização para o *step* 3 para otimizar o vetor de saída do melhor modelo do *step* 1, utilizando como função de aptidão (do inglês *fitness*) o melhor modelo do *step* 2;
- Propor uma arquitetura final de execução para predição e otimização de cenários promotores de eficiência no consumo de energia da Unidade Operativa de Pintura.

1.6 Organização da Dissertação

A estrutura deste trabalho é apresentada de maneira organizada e sequencial. A seguir, detalhamos cada parte dessa estrutura para facilitar a compreensão:

1. Capítulo 1: Contexto e Introdução

- Breve introdução contextualizando o trabalho.
- Revisão de trabalhos relacionados para situar o leitor no campo de estudo.
- Explicação das motivações que impulsionaram a realização deste trabalho.
- Definição clara dos objetivos a serem alcançados.

2. Capítulo 2: Fundamentação Teórica

- Apresentação dos principais conceitos e teorias fundamentais relacionados aos modelos utilizados neste estudo.
- Explicação detalhada dos modelos teóricos e sua relevância para a pesquisa.

3. Capítulo 3: Metodologia

- Descrição da metodologia empregada na condução deste trabalho, descrevendo a arquitetura de 3 *steps* proposta.
- Detalhamento dos procedimentos, técnicas e ferramentas.

4. Capítulo 4: Resultados e Discussão

- Apresentação dos resultados obtidos a partir da aplicação da metodologia.
- Análise crítica e discussão dos resultados em relação aos objetivos propostos.
- Definição da arquitetura final.

5. Capítulo 5: Considerações Finais e Trabalhos Futuros

- Recapitulação das descobertas e conclusões alcançadas.
- Proposição de trabalhos futuros que possam surgir a partir das lacunas identificadas ou dos resultados obtidos neste estudo.

2 Fundamentação Teórica

Essa fundamentação teórica visa fornecer uma base sólida para identificar o estado da arte da análise do consumo energético em oficinas de pintura automotiva e apresentar as técnicas utilizadas na abordagem metodológica. Compreender o processo de manufatura industrial é essencial para abordar os desafios relacionados à Unidade Operativa de Pintura.

2.1 Área do Negócio

A Manufatura Industrial abrange um conjunto de atividades e processos intrínseco à produção de bens e produtos a partir da utilização de diversos vetores energéticos, como eletricidade, combustíveis, vapor, entre outros. Esses vetores energéticos alimentam máquinas, equipamentos e ferramentas utilizadas ao longo da linha de produção. A Manufatura Industrial está também diretamente ligada à automação e à utilização de tecnologias avançadas, como robótica, inteligência artificial e sistemas integrados de produção. Essas inovações têm o objetivo de otimizar os processos, aumentar a produtividade, reduzir custos e melhorar a qualidade dos produtos finais.

Entre as unidades que fazem parte das etapas produtivas, conforme Figura 1, este trabalho tem como objeto de estudo a Unidade Operacional da Pintura. Como fluxo de produção, a área da Pintura (ou Oficina de Pintura) tem por responsabilidade receber as carrocerias montadas e soldadas sem os componentes finais do processo que lhe antecede, a Funilaria. As carrocerias são recebidas a partir de transportadores (esteira de produção) em um fluxo contínuo. Após passar por todos os processos, as carrocerias seguem na esteira para a próxima etapa, a unidade de Montagem, que recebe um produto limpo de impurezas, quimicamente tratada e com aplicação de toda sua pintura.

No contexto das organizações de manufatura, busca-se constantemente adotar uma ampla gama de técnicas, estratégias e ferramentas para alcançar e superar seus objetivos fundamentais. Essas estratégias e ferramentas vão desde a gestão eficiente de recursos humanos e materiais até a implementação de tecnologias avançadas e processos inovadores. A aplicação de tecnologias emergentes, como Internet das Coisas (do inglês *Internet of Things* - IoT), *Big Data Analytics*, Inteligência Artificial (IA) e Manufatura Aditiva (impressão 3D) tem desempenhado papéis relevantes na modernização e melhoria contínua dos processos de manufatura. Essas tecnologias permitem monitorar em tempo real o desempenho das máquinas, prever falhas e tomar decisões baseadas em dados para otimizar a produtividade e a qualidade dos produtos.

Nesse contexto, a adoção e implementação de técnicas avançadas como ML, DL

e mineração de dados emergem como estratégias altamente benéficas para enfrentar e superar os desafios complexos que a indústria de fabricação enfrenta atualmente[6].

ML é uma subárea da inteligência artificial que oferece a capacidade de analisar e interpretar grandes volumes de dados de forma automatizada, identificando padrões e tendências significativas que seriam difíceis de perceber por métodos tradicionais. Isso permite que as organizações de manufatura tomem decisões mais informadas e baseadas em dados precisos, melhorando a eficiência operacional, a qualidade dos produtos e a satisfação do cliente.

DL é uma subárea da inteligência artificial que se baseia em redes neurais artificiais compostas por múltiplas camadas de processamento, permitindo a aprendizagem de representações complexas de dados. Essas redes são capazes de aprender automaticamente a partir dos dados, identificando padrões e realizando tarefas como reconhecimento de padrões, classificação, detecção de objetos e tradução de idiomas. O termo "profundo" refere-se ao número de camadas que compõem a rede, possibilitando a extração de características de alto nível em dados de entrada.

A mineração de dados envolve a extração e análise de informações a partir de conjuntos de dados complexos, fornecendo uma compreensão mais profunda dos processos de fabricação, identificando áreas de melhoria, detectando anomalias e otimizando o desempenho geral da linha produtiva.

A combinação dessas técnicas permite a implementação de sistemas de manufatura inteligentes e adaptativos, capazes de aprender e se ajustar automaticamente às mudanças nas demandas do mercado, às variações nos processos de produção e às necessidades específicas dos clientes. Isso resulta em uma operação mais eficiente, redução de custos, menor tempo de inatividade das máquinas e maior agilidade para responder às demandas do mercado em constante evolução (JIANG et al., 2023).

2.2 Modelos de Inteligência Computacional

Modelos de ML, Redes neurais, algoritmos genéticos, e outros modelos exibem eficácia na análise de extensos conjuntos de dados e na identificação de padrões complexos. Quando aplicados na gestão energética industrial, esses modelos viabilizam a antecipação de padrões de consumo, a otimização da distribuição de energia e a adaptação das operações para mitigar picos de demanda proporcionando o desenvolvimento de sistemas inteligentes de monitoramento e controle, que ajustam os parâmetros operacionais com o intuito de assegurar um consumo energético eficiente.

2.2.1 *Random Forest*

A RF é uma técnica de aprendizado de máquina proposta por Breiman (2001). Ela faz parte do grupo de modelos ensembles. Ensemble refere-se a técnicas que combinam múltiplos modelos de aprendizado de máquina para produzir previsões mais robustas e precisas do que os modelos individuais. A RF consiste em um conjunto de árvores de decisão, cada uma treinada em um subconjunto aleatório dos dados de treinamento.

Uma árvore de decisão é uma estrutura hierárquica composta por nós e folhas. Cada nó representa uma decisão baseada em uma característica e um valor de corte, enquanto as folhas representam as previsões finais. A construção de uma árvore envolve a escolha da melhor característica para divisão em cada nó, com base em critérios como a impureza dos dados medida pela entropia ou pelo índice Gini para problemas de classificação e erro quadrático médio (do Inglês *Mean Square Error* - MSE) ou o erro absoluto médio (do inglês *Mean Absolute Error* - MAE) para regressão.

Este trabalho aborda problemas de regressão, portanto, a medida de impureza adotada para construção dos nós é o *MSE*, dado pela equação 2.1. Após treinar um conjunto de árvores de decisão, a RF realiza as previsões combinando as previsões de todas as árvores por meio da média para produzir uma estimativa final.

$$MSE(S) = \frac{1}{|S|} \sum_{i \in S} (y_i - \bar{y})^2 \quad (2.1)$$

Onde S é um dos *splits* resultantes do ponto de corte, $|S|$ é o número de exemplos de S , y_i é o valor real do conjunto S e \bar{y} é a média dos valores reais do conjunto S .

A aleatoriedade na RF é introduzida de duas maneiras:

- Amostragem de Dados (do inglês *Bagging*): Cada árvore na floresta é treinada em um subconjunto aleatório dos dados de treinamento, selecionado com substituição. Isso ajuda a reduzir o sobreajuste, pois cada árvore vê apenas uma parte dos dados durante o treinamento, e aumenta a diversidade dos modelos na floresta.
- Amostragem de Características: Em cada divisão de nó durante a construção de cada árvore, apenas um subconjunto aleatório das características é considerado. Isso introduz ainda mais diversidade entre as árvores na floresta e ajuda a evitar que características dominantes dominem o processo de divisão, tornando o modelo mais robusto.

2.2.2 *Extreme Gradient Boosting*

O Aumento Extremo de Gradiente (do inglês *Extreme Gradient Boosting* - XGBoost) é um algoritmo de ML desenvolvido por Chen e Guestrin (2016). É um modelo ensemble

baseado no conceito de boosting, onde vários modelos de aprendizado fracos são combinados para formar um modelo forte. O XGBoost opera de forma iterativa, construindo árvores de decisão sequencialmente para melhorar o desempenho do modelo.

Conforme apresentado por Zhu et al. (2024), a previsão é calculada agregando os resultados de cada modelo, conforme demonstrado na Equação 2.2:

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i), \quad (2.2)$$

onde i varia de 1 a n . Aqui, n representa o número de amostras, enquanto f_k denota a k -ésima função da árvore de regressão.

A função objetivo é dada por uma função de perda $L(y_i, \hat{y}_i)$ e um termo de regularização $\Omega(f_i)$, conforme mostrado na Equação 2.3:

$$f_{obj} = \sum_{i=1}^n L(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k), \quad (2.3)$$

No tempo t , a função objetivo pode ser representada pela Equação 2.4:

$$f(t)_{obj} = \sum_{i=1}^n L(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) + C \quad (2.4)$$

O método de expansão de Taylor é utilizado pelo XGBoost para aproximar a função objetivo, como indicado na Equação 2.5:

$$f(t)_{obj} \approx \sum_{i=1}^n (L(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)) + \Omega(f_t) + C, \quad (2.5)$$

onde g_i e h_i são definidos como:

$$g_i = \frac{\partial L(y, f_{t-1}(\vec{X}))}{\partial f_{t-1}(\vec{X})} \quad (2.6)$$

$$h_i = \frac{\partial^2 L(y, f_{t-1}(\vec{X}))}{\partial^2 f_{t-1}(\vec{X})} \quad (2.7)$$

O resultado previsto no tempo t é dado por:

$$f_t(x) = w_{q(x)}, w \in R^T, q : R^d \rightarrow \{1, 2, \dots, T\}, \quad (2.8)$$

onde q representa a estrutura da árvore e w é o peso da folha da árvore.

O termo de regularização $\Omega(f_i)$ representa a complexidade do modelo para evitar o excesso de ajuste, definido na Equação 2.9:

$$\Omega(f_t) = \gamma T + \frac{1}{2} \lambda \|w\|^2, \quad (2.9)$$

onde T é o número de nós folha, w é a soma dos *escores* dos nós folha nas árvores, e γ, λ são parâmetros ajustados.

Dessa forma, a função objetivo pode ser expressa como:

$$f_{obj}^{(t)} = \sum_{j=1}^T \left[\left(\sum_{i \in I_j} g_i \right) w_j + \frac{1}{2} \left(\sum_{i \in I_j} h_i + \lambda \right) w_j^2 \right] + \gamma T, \quad (2.10)$$

onde I_j representa o conjunto de amostras folha na j -ésima árvore.

Em seguida, definimos a função objetivo $G_j = \sum_{i \in I_j} g_i$ e $H_j = \sum_{i \in I_j} h_i$ que pode ser simplificada da seguinte forma:

$$f_{obj} = -\frac{1}{2} \sum_{j=1}^T \frac{G_j}{H_j + \lambda} + \gamma T. \quad (2.11)$$

A função de perda no XGBoost é uma combinação da perda do modelo e da penalidade de regularização. Para problemas de classificação binária, a perda do modelo pode ser a função logística, enquanto para problemas de regressão, pode ser *MSE* (Equação 2.1). A penalidade de regularização é controlada por parâmetros como a profundidade máxima da árvore (*max_depth*) e o coeficiente de regularização (λ). Durante a construção de cada árvore, o algoritmo busca a divisão ótima em cada nó, considerando a redução máxima na função de perda.

2.2.3 Redes Neurais Artificiais

O conceito de DL surgiu para descrever redes neurais com múltiplas camadas ocultas. Essas redes têm a capacidade de aprender representações complexas de dados, o que lhes permite realizar tarefas como previsões baseadas em séries temporais (HEWAMALAGE; BERGMEIR; BANDARA, 2021), processamento de linguagem natural (ZONG; KRISHNAMACHARI, 2022), visão computacional (BHARATI; PRAMANIK, 2020).

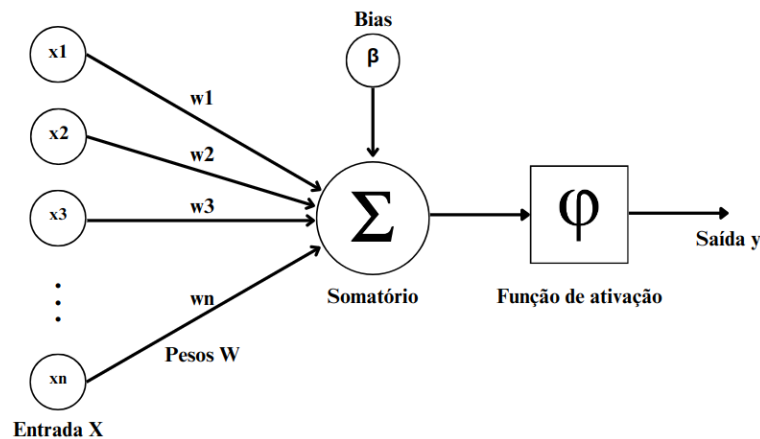
McCulloch e Pitts (1943) propuseram o primeiro modelo simplificado de um neurônio artificial, conhecido como o modelo McCulloch-Pitts. Este modelo inspirou o desenvolvimento inicial de redes neurais. Rosenblatt (1958) propôs o Perceptron, um modelo de neurônio artificial com capacidade de aprendizado supervisionado. O Perceptron foi uma das primeiras formas de ML. Rumelhart, Hinton e Williams (1986) demonstraram a eficácia do algoritmo *backpropagation* para treinamento de redes neurais. O trabalho de LeCun, Bengio e Hinton (2015) foi um dos pioneiros em redes neurais convolucionais (do inglês *Convolutional Neural Network* - CNN), contribuindo significativamente para o desenvolvimento de algoritmos de DL para reconhecimento de padrões em imagens. As redes neurais no campo do DL continuaram avançando, impulsionadas por pesquisadores, inovações tecnológicas e aplicações práticas em diversos domínios.

O funcionamento de um neurônio artificial é inspirado no neurônio biológico. Um neurônio artificial baseado no modelo Perceptron recebe um conjunto de entradas ponderadas, aplica uma função de ativação a essa soma ponderada e produz uma saída. Matematicamente, podemos representar isso pela Equação 2.12 e sua estrutura é observada na Figura 2.

$$y = f\left(\sum_{i=1}^n w_i \cdot x_i + b\right), \quad (2.12)$$

onde y é a saída do neurônio, x_i são os valores de entrada, w_i são os pesos associados a cada entrada, b é o *bias* e f é a função de ativação.

Figura 2. Neurônio Artificial Perceptron.

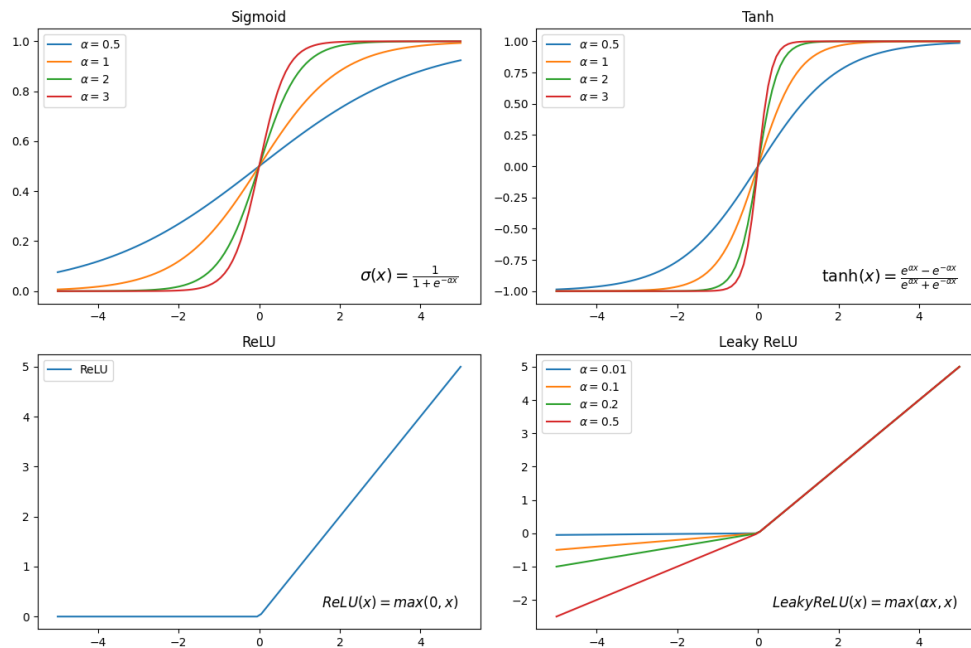


Fonte: os autores

As funções de ativação desempenham um papel fundamental nas redes neurais, introduzindo não linearidades nas saídas das unidades neurais e permitindo que as redes aprendam relações complexas nos dados (APICELLA et al., 2021). A função sigmóide (σ), também conhecida como função logística, é uma das funções de ativação mais comuns. Produz saídas no intervalo de 0 a 1 e é útil para problemas de classificação binária devido à sua interpretação probabilística das saídas. A função tangente hiperbólica (\tanh) produz saídas no intervalo de -1 a 1 e é útil para normalizar saídas de camadas intermediárias. A Unidade Linear Retificada (do inglês *Rectified Linear Unit - ReLU*) atribui zero para valores negativos e mantém os valores positivos intactos. É amplamente adotada devido à sua eficácia na mitigação do problema de desvanecimento do gradiente (HU et al., 2018). Uma variação da função *ReLU* é a Unidade Linear Retificada Inclinada (do inglês *Inclined Rectified Linear Unit - LeakyReLU*), que introduz uma inclinação para valores negativos, evitando neurônios inativos e permitindo um fluxo de gradiente contínuo.

Essas são algumas das principais funções de ativação utilizadas em redes neurais. A escolha da função de ativação depende do problema específico em questão e da arquitetura

Figura 3. Curvas das funções σ , ReLU, \tanh e $LeakyReLU$.



Fonte: os autores

da rede neural. A Figura 3 apresenta as curvas das quatro funções descritas acima. Para as funções $LeakyReLU$, $sigmoide$ e \tanh , três curvas foram plotadas variando o parâmetro α .

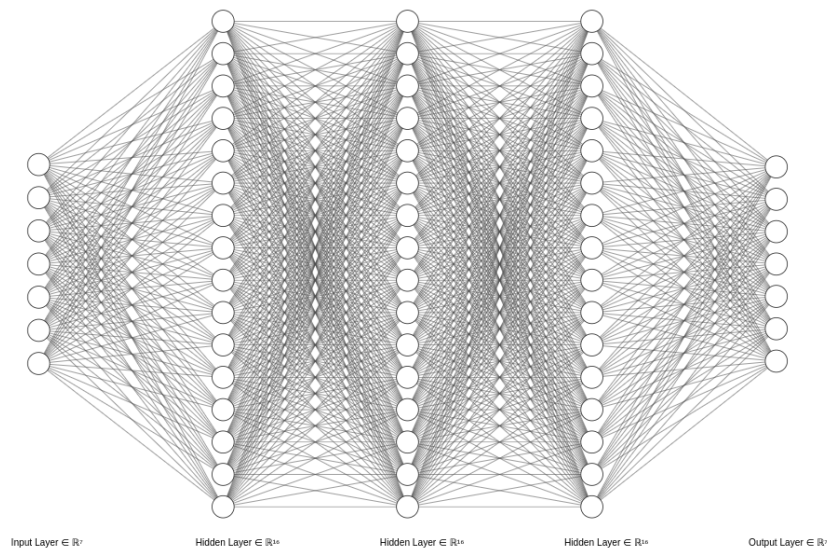
Uma rede neural é descrita em 3 grupos de camadas: camada de entrada, camadas ocultas e camada de saída.

- Camada de entrada (do inglês *input layer*): é a primeira camada da rede neural e recebe os dados de entrada do problema. Cada nó (ou neurônio) nesta camada representa uma característica ou uma entrada do conjunto de dados. Não há processamento realizado nesta camada, ela apenas passa os dados de entrada para a próxima camada (camada oculta).
- Camada oculta (do inglês *hidden layer*): As camadas ocultas estão localizadas entre a camada de entrada e a camada de saída. Cada camada oculta consiste em um conjunto de neurônios interconectados. Nessa camada, cada neurônio recebe entradas dos neurônios da camada anterior, realiza uma transformação linear e aplica uma função de ativação para produzir uma saída.
- Camada de saída (do inglês *output layer*): A camada de saída é a última camada da rede neural e produz os resultados finais ou as previsões do modelo. O número de neurônios na camada de saída depende do tipo de tarefa que a rede neural está realizando. Por exemplo, em um problema de classificação binária, pode haver um

único neurônio na camada de saída para representar a probabilidade de pertencer a uma classe.

As redes neurais profundas ou densas (RNA Densa) são caracterizadas por terem várias camadas ocultas, conforme o exemplo na Figura 4, que descreve a estrutura de uma rede com 7 neurônios na camada de entrada, 3 camadas ocultas com 16 neurônios em cada camada e a camada de saída com 7 neurônios. Essas redes são capazes de aprender representações hierárquicas de dados, o que as torna eficazes em tarefas complexas de ML.

Figura 4. Exemplo de estrutura de uma rede neural densa de 5 camadas.



Fonte: os autores

O treinamento de redes neurais consiste em ajustar os parâmetros do modelo para minimizar uma função de custo (função de perda). Este processo geralmente envolve algoritmos de otimização, como o gradiente descendente, que atualizam iterativamente os parâmetros do modelo com base no gradiente da função de custo em relação aos parâmetros. O treinamento pode ser descrito através dos seguintes passos:

- Inicialização dos pesos: Os pesos da rede neural são inicializados aleatoriamente para evitar que o modelo comece em um estado indesejável. Isso pode ser feito, por exemplo, gerando os pesos de uma distribuição normal com média zero e desvio padrão pequeno.
- Propagação direta (do inglês *forward propagation*): Durante esta etapa, os dados de entrada são alimentados na rede neural e passam por uma série de camadas ocultas até chegar à camada de saída. Cada camada realiza uma transformação linear dos dados seguida por uma função de ativação não linear. A saída de uma camada l é calculada da seguinte forma:

$$z^{(l)} = W^{(l)} \cdot a^{(l-1)} + b^{(l)} \quad (2.13)$$

$$a^{(l)} = g^{(l)}(z^{(l)}) \quad (2.14)$$

Onde $W^{(l)}$ é a matriz de pesos da camada l , $a^{(l-1)}$ é a saída da camada anterior, $b^{(l)}$ é o vetor de viés da camada l , $g^{(l)}$ é a função de ativação da camada l , $z^{(l)}$ é a entrada ponderada da camada l , e $a^{(l)}$ é a saída da camada l .

- Cálculo da função de custo (do inglês *loss function*): Após a propagação direta, a saída predita da rede neural é comparada com as saídas reais dos dados de treinamento. A função de custo, que quantifica o quão boa é a previsão do modelo em relação aos rótulos reais, é calculada. Seja \hat{y} a saída prevista da rede neural e y o rótulo verdadeiro, a função de custo J pode ser definida como:

$$J = \frac{1}{m} \sum_{i=1}^m L(\hat{y}_i, y_i) \quad (2.15)$$

onde L é uma função de custo individual, como a entropia cruzada para classificação ou o *MSE*, e m é o número de exemplos de treinamento.

- Propagação reversa (do inglês *backpropagation*): O objetivo desta etapa é calcular o gradiente da função de custo em relação aos parâmetros do modelo, para que possamos ajustá-los de forma a minimizar a função de custo. O algoritmo *backpropagation* calcula esses gradientes recursivamente, começando pela última camada e se movendo de volta para as camadas acultas da rede.
- Atualização dos pesos: Com os gradientes calculados, podemos atualizar iterativamente os pesos da rede neural na direção que minimiza a função de custo. A regra de atualização dos pesos é geralmente dada por:

$$w_{ij}^{(l)} = w_{ij}^{(l)} - \alpha \frac{\partial J}{\partial w_{ij}^{(l)}} \quad (2.16)$$

onde $w_{ij}^{(l)}$ é o peso da conexão entre a unidade i na camada $l - 1$ e a unidade j na camada l , e α é a taxa de aprendizado, que controla a magnitude das atualizações de peso.

À medida que o treinamento avança, espera-se que a função de custo diminua, indicando que o modelo está se ajustando aos dados de treinamento e melhorando suas previsões. No entanto, é importante monitorar também o desempenho do modelo em um conjunto de dados de validação separado para evitar *overfitting*, ou seja, para garantir que o modelo esteja generalizando bem para dados não vistos.

2.2.4 Long Short Term Memory

LSTM foi introduzida pela primeira vez por Hochreiter e Schmidhuber (1997). É um tipo especializado de Rede Neural Recorrente (do inglês *Recurrent Neural Network* - RNN) desenvolvida para lidar com sequências de dados e capturar dependências temporais de longo prazo. Uma RNN, conforme descrita por Pascanu et al. (2013), representa um modelo que emula um sistema dinâmico discreto ao longo do tempo, com uma entrada x_t , uma saída y_t e um estado oculto h_t para cada instante t . Esse sistema dinâmico é caracterizado por duas funções principais: uma função de transição de estado f_h e uma função de saída f_o , ambas parametrizadas por conjuntos específicos de parâmetros θ_h e θ_o .

Em cada etapa de tempo t , o estado oculto h_t é atualizado com base na entrada atual x_t e no estado oculto anterior h_{t-1} , seguindo a equação:

$$h_t = f_h(x_t, h_{t-1}) \quad (2.17)$$

Posteriormente, a saída y_t é computada com base no estado oculto h_t , de acordo com a função de saída f_o :

$$y_t = f_o(h_t) \quad (2.18)$$

Para o treinamento efetivo do modelo RNN, a partir de um conjunto de N sequências de treinamento, os parâmetros da rede podem ser estimados pela minimização de uma função de custo $J(\theta)$. Essa função de custo é definida como a média de uma medida de divergência predefinida $d(y_t^{(n)}, f_o(h_t^{(n)}))$ entre a saída real $y_t^{(n)}$ e a saída prevista $f_o(h_t^{(n)})$, para cada etapa de tempo t e cada sequência n . O estado oculto inicial $h_0^{(n)}$ é comumente fixado em zero.

A função de custo $J(\theta)$ é expressa como:

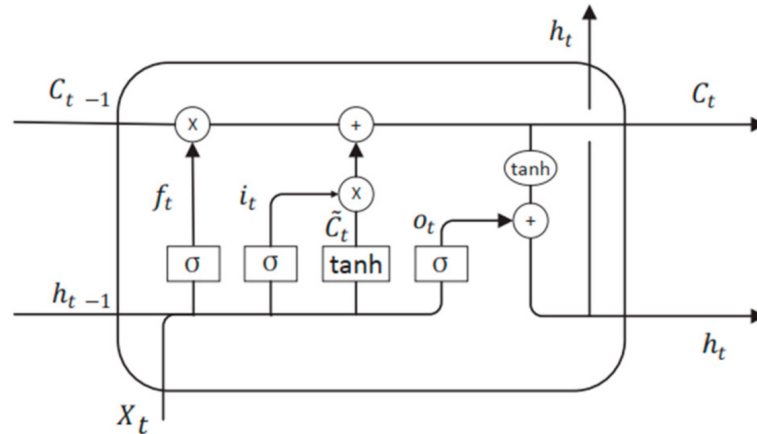
$$J(\theta) = \frac{1}{N} \sum_{n=1}^N \sum_{t=1}^{T_n} d(y_t^{(n)}, f_o(h_t^{(n)})) \quad (2.19)$$

onde $h_t^{(n)} = f_h(x_t^{(n)}, h_{t-1}^{(n)})$ e $h_0^{(n)} = 0$.

Em comparação com redes neurais tradicionais, as LSTM's possuem uma arquitetura mais complexa, incorporando unidades de memória interna chamadas de células. Essas células são capazes de armazenar informações ao longo do tempo e controlar o fluxo de informações na rede, permitindo que as LSTMs mantenham e atualizem estados de memória por longos períodos.

Uma célula LSTM os seguintes componentes: *input gate* i_t , *forget gate* f_t , *output gate* o_t , estado de célula candidata \tilde{c} , estado de célula c_t e um estado oculto h_t , dado pelas

Figura 5. Arquitetura da LSTM.



Fonte: Bouktif et al. (2020)

Equações 2.20, 2.21, 2.22, 2.23, 2.24 e 2.25, respectivamente. A função do *input gate* é determinar quais informações devem ser armazenadas no estado de célula atual, enquanto o *forget gate* identifica as informações que devem ser descartadas. Por fim, o *output gate* determina quais informações serão transmitidas para o estado oculto atual (TELILI et al., 2023).

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.20)$$

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.21)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (2.22)$$

$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (2.23)$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \quad (2.24)$$

$$h_t = o_t \cdot \tanh(c_t) \quad (2.25)$$

Onde W_x e b_x representam a matriz de pesos e o vetor de bias de cada equação, respectivamente.

2.2.5 Gated Recurrent Unit

A rede Unidade Recorrente Fechada (do inglês *Gated Recurrent Unit* - GRU), proposta por Cho et al. (2014), é uma arquitetura que pertence à família das RNNs. Assim como a LSTM, a GRU é projetada para lidar com problemas de sequência, onde a entrada ou saída é uma sequência de dados, como em tradução de texto, previsão de séries temporais, entre outros.

Conforme mostrado na Figura 6, a GRU possui duas portas principais: *update gate* e *reset gate*. A *update gate* controla quanto do estado anterior deve ser mantido e quanto do novo estado proposto deve ser adicionado. Enquanto isso, a *reset gate* decide quanto do estado anterior deve ser esquecido. Essas portas permitem que a GRU aprenda a memorizar informações importantes de maneira mais eficiente do que as RNNs tradicionais.

O *update gate* Z_t é calculado da seguinte forma:

$$Z_t = \sigma(W(Z)x_t + U(Z)h_{t-1}) \quad (2.26)$$

O *reset gate* r_t é calculado da seguinte forma:

$$r_t = \sigma(W(r)x_t + U(r)h_{t-1}) \quad (2.27)$$

O conteúdo atual da memória h'_t é calculado da seguinte forma:

$$h'_t = \tanh(Wx_t + r_t \odot Uh_{t-1}) \quad (2.28)$$

A memória final no momento atual h_t é calculada da seguinte forma:

$$h_t = Z_t \odot h_{t-1} + (1 - Z_t) \odot h'_t \quad (2.29)$$

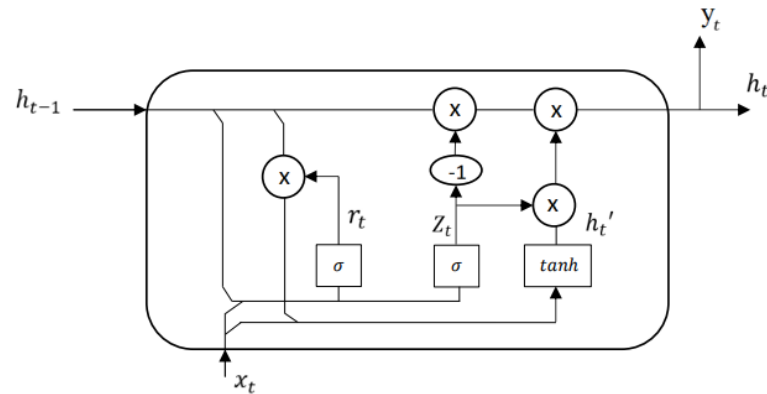
onde; x_t é o vetor de entrada, h_{t-1} é a saída anterior, W e U são os pesos, σ é a função sigmoide, e \odot é o produto de Hadamard.

As GRUs tendem a ser mais fáceis de treinar e mais eficientes computacionalmente do que as RNNs tradicionais. Isso se deve ao fato de que as GRUs possuem menos parâmetros, o que reduz a complexidade de configuração do modelo e a torna mais fácil de otimizar em cada etapa de execução durante o treinamento. Essa eficiência computacional é especialmente importante em aplicações onde o tempo de treinamento e a capacidade de processamento são limitados.

2.2.6 Modelo Híbrido GRU-LSTM

À medida que a complexidade da modelagem de sequências aumenta, surge a necessidade de abordagens mais eficazes. Entre as técnicas amplamente exploradas estão as RNNs, e suas variantes GRU e LSTM. Embora essas arquiteturas tenham demonstrado

Figura 6. Arquitetura da GRU.

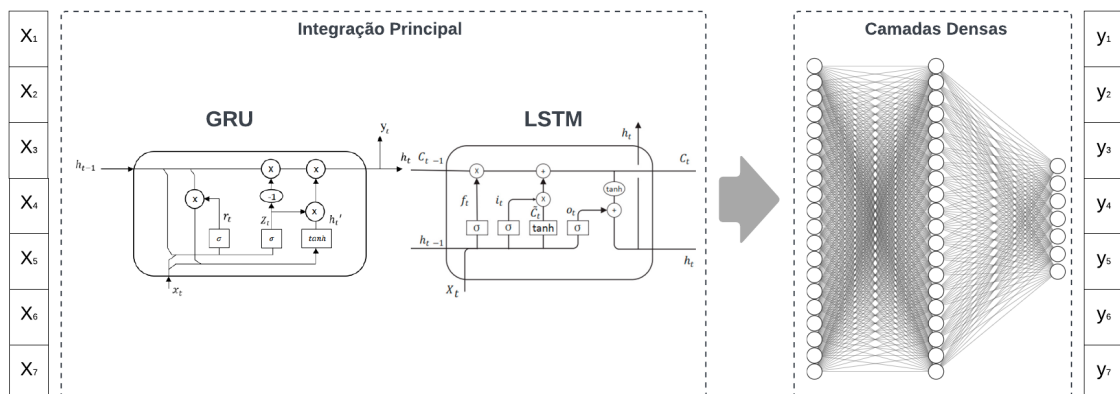


Fonte: Ragb et al. (2023)

sucesso em capturar dependências temporais em dados sequenciais, cada uma possui suas próprias limitações: as GRUs são conhecidas por sua eficiência computacional, mas podem não capturar dependências de longo prazo tão bem quanto as LSTMs, que, por sua vez, podem ser mais custosas computacionalmente (YANG; YU; ZHOU, 2020; CAHUANTZI; CHEN; GÜTTEL, 2023).

Uma série de estudos tem explorado o uso de modelos híbridos GRU-LSTM em várias aplicações. Elsayed, Maida e Bayoumi (2019) descobriu que um modelo híbrido de rede totalmente convolucional GRU superou o estado da arte na classificação de séries temporais. Patra e Mohanty (2022) aplicou um modelo híbrido LSTM-GRU à previsão de preços de ações, obtendo resultados mais precisos do que os modelos autônomos LSTM, GRU e MLP. Asl *et al.* (2023) demonstrou a superioridade de um modelo híbrido LSTM-GRU na previsão de pontuação de crédito, superando os métodos tradicionais e os modelos autônomos LSTM e GRU.

Figura 7. Arquitetura do modelo híbrido.



Fonte: os autores

Esses estudos sugerem, em conjunto, que os modelos híbridos GRU-LSTM podem

oferecer melhor desempenho em uma série de aplicações. Portanto, este trabalho buscou implementar um modelo híbrido GRU-LSTM, visando aproveitar a eficiência computacional das GRUs e a capacidade de memorização de longo prazo da LSTMs para enfrentar desafios complexos na modelagem de séries temporais.

Figura 8. Pseudocódigo do Modelo Híbrido GRU-LSTM

Algoritmo 1: Pseudocódigo do Modelo Híbrido GRU-LSTM

Dados: Sequência de entrada X de formato $(lag, Feature)$

Resultado: Saída predita pela rede

Inicialize os pesos das camadas:

$W_{GRU}, b_{GRU};$

$W_{LSTM}, b_{LSTM};$

$W_{Dense1}, b_{Dense1};$

$W_{Dense2}, b_{Dense2};$

$W_{out}, b_{out};$

enquanto Critério de parada não for atingido **faça**

$GRU_{output} = GRU_{layer}(X, W_{GRU}, b_{GRU});$

$LSTM_{output} = LSTM_{layer}(GRU_{output}, W_{LSTM}, b_{LSTM});$

$Dense1_{output} = Dense_{layer}(LSTM_{output}, W_{Dense1}, b_{Dense1});$

$Dense2_{output} = Dense_{layer}(Dense1_{output}, W_{Dense2}, b_{Dense2});$

$output = Dense2_{output} \times W_{out} + b_{out};$

 Validação do Modelo;

 Atualização dos pesos;

fim

retorna $output$

Fonte: Autores

O pseudocódigo da Figura 8 oferece uma visão macro de execução do modelo híbrido, da mesma forma, a Figura 7 mostra a arquitetura completa deste modelo, descrevendo um modelo híbrido baseado em DL que combina camadas GRU e LSTM para a previsão de sequências em um problema de aprendizado de máquina. Ele inicia com a inicialização dos pesos das camadas do modelo. Em cada iteração, as entradas são passadas pelas camadas GRU e LSTM, gerando saídas intermediárias que são processadas por camadas densas adicionais. O modelo é treinado iterativamente até que um critério de parada seja atingido, como um número máximo de épocas de treinamento ou a convergência para um desempenho satisfatório na validação do modelo.

2.2.7 Genetic Algorithm

O Algoritmo Genético (do inglês *Genetic Algorithm* - GA), proposto por Holland (1975), pertence a uma classe de algoritmos de otimização inspirados na teoria da evolução de Darwin. Sua execução é apresentada na Figura 9, ele opera com uma população de soluções candidatas, representadas por cromossomos ou indivíduos, que são submetidos a

um conjunto de operadores genéticos para evoluir ao longo das gerações em direção a uma solução ótima ou próxima dela (KATOCH; CHAUHAN; KUMAR, 2021). O algoritmo conta com uma função de avaliação que atribui um valor de aptidão a cada solução candidata com base em seu desempenho no problema de otimização.

O objetivo é maximizar ou minimizar o valor da função de aptidão, dependendo da natureza do problema. Os operadores genéticos são componentes-chave do GA. Existem três operadores genéticos: seleção, cruzamento e mutação.

- **Seleção:** Durante a seleção, os indivíduos da população são selecionados com base em sua aptidão para reprodução. Diversos métodos de seleção podem ser utilizados, como a roleta, torneio e seleção por classificação, cada um com suas vantagens e desvantagens.
- **Cruzamento:** Processo pelo qual os indivíduos selecionados são combinados para gerar descendentes. Isso simula a recombinação genética na natureza e ajuda a explorar novas regiões do espaço de busca. Operadores comuns incluem o cruzamento de um ponto, cruzamento de dois pontos e cruzamento uniforme.
- **Mutação:** A mutação é um operador que introduz pequenas alterações aleatórias nos indivíduos descendentes para aumentar a diversidade genética da população. Isso ajuda a evitar a convergência prematura para ótimos locais subótimos e a explorar mais completamente o espaço de busca.

Figura 9. Pseudocódigo do GA

Algoritmo 2: Pseudocódigo do GA

Dados: Tamanho da população (P), Taxa de cruzamento (C_R)

Resultado: Melhor cromossomo encontrado na população final

Inicialize a população com cromossomos aleatórios;

Avalie o fitness de cada cromossomo na população;

enquanto *Critério de parada não for atingido* **faça**

 Selecione indivíduos para reprodução;

 Realize o cruzamento para gerar descendentes;

 Aplique mutação aos descendentes;

 Avalie o fitness dos descendentes;

 Substitua a população atual pelos descendentes;

se *Elitismo* **então**

 | Aplique elitismo;

fim

fim

Retorne o melhor cromossomo encontrado na população final;

O GA conta também com a estratégia de elitismo para garantir que os indivíduos mais aptos de uma população atual sejam preservados para a próxima geração, mesmo que não sejam selecionados para reprodução. Ao implementar o elitismo, uma certa porcentagem dos melhores indivíduos da população atual é automaticamente transferida para a próxima geração. Isso garante que as características genéticas benéficas presentes nesses indivíduos sejam mantidas e transmitidas para as gerações futuras, ajudando a preservar a qualidade global da população. O pseudocódigo da figura 9 descreve o processo iterativo do GA.

2.2.8 Particle Swarm Optimization

O algoritmo Otimização por Enxame de Partículas (do inglês *Particle Swarm Optimization* - PSO) é uma técnica de otimização computacional baseada em simulação de enxame, inspirada no comportamento de pássaros e cardumes de peixes. A ideia central do PSO é simular o comportamento de um grupo de partículas, cada uma representando uma solução candidata para o problema de otimização.

Figura 10. Pseudocódigo do PSO

Algoritmo 3: Pseudocódigo do PSO

Dados: Tamanho da população (P), coeficientes w , c_1 e c_2
Resultado: Retorne a melhor posição encontrada pelas partículas
 Inicialize as partículas com posições e velocidades aleatórias;
enquanto *Critério de parada não for atingido* **faça**
 para cada *cada partícula* **faça**
 Avalie o valor da função de aptidão para a posição atual da partícula;
 se *Valor da função de aptidão é melhor que o melhor valor pessoal da partícula* **então**
 | Atualize o melhor valor pessoal da partícula;
 fim
 se *Valor da função de aptidão é melhor que o melhor valor global* **então**
 | Atualize o melhor valor global;
 fim
 Atualize a velocidade e a posição da partícula;
 fim
fim
 Retorne a melhor posição encontrada pelas partículas;

Fonte: Os autores

A Figura 10 descreve o pseudocódigo do PSO, onde um grupo de partículas (representando soluções potenciais) se move pelo espaço de busca em busca da melhor solução. No início, as partículas são inicializadas com posições e velocidades aleatórias. Em cada iteração, a função de aptidão de cada partícula é avaliada, e as partículas atualizam

suas velocidades e posições de acordo com regras que levam em consideração seu próprio desempenho e o desempenho global do enxame. O processo continua até que um critério de parada seja atingido, como um número máximo de iterações ou a convergência para uma solução satisfatória (WANG et al., 2018).

A atualização da velocidade $v_{id}(t+1)$ de uma partícula i de dimensão d , na iteração $t+1$ é calculada pela equação 9 e sua posição $x_{id}(t+1)$ é dada pela equação 10.

$$L(t) = c_1 \cdot r_1 \cdot (p_{id} - x_{id}(t)) \quad (2.30)$$

$$G(t) = c_2 \cdot r_2 \cdot (p_{gd} - x_{id}(t)) \quad (2.31)$$

$$v_{id}(t+1) = w \cdot v_{id}(t) + L(t) + G(t) \quad (2.32)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (2.33)$$

Onde w é o coeficiente de inércia, c_1 e c_2 são os coeficientes que controlam a influência do melhor valor pessoal e global, respectivamente, r_1 e r_2 são números aleatórios entre 0 e 1, p_{id} e p_{gd} são as melhores posições já alcançadas pela partícula i e pelo enxame até o momento, respectivamente. $v_{id}(t)$ e $x_{id}(t)$ descrevem a velocidade e a posição da partícula i na dimensão d na iteração atual t , respectivamente. Sua execução pode ser observada através do pseudocódigo 3.

2.2.9 Differential Evolution

O algoritmo Evolução Diferencial (do inglês *Differential Evolution* - DE) é um método de otimização baseado em população, inspirado no processo de evolução natural. Segundo Opara e Arabas (2019), uma população de vetores, onde cada vetor representa uma solução candidata para o problema de otimização em questão. De acordo com o pseudocódigo na Figura 11, o algoritmo começa gerando aleatoriamente uma população inicial de soluções (vetores) candidatas dentro do espaço de busca. Cada vetor passa por uma mutação diferencial, onde um vetor mutante é gerado combinando diferenças ponderadas entre três vetores selecionados aleatoriamente da população. A mutação é realizada pela seguinte equação:

$$v_i = x_{r1} + F \cdot (x_{r2} - x_{r3}) \quad (2.34)$$

Onde v_i é o novo vetor mutante gerado para o indivíduo i na geração g ; x_{r1} , x_{r2} e x_{r3} são três vetores distintos selecionados aleatoriamente da população atual, exceto

o próprio indivíduo i . F é um parâmetro de controle chamado de fator de escala, que controla o grau de perturbação aplicado à diferença entre os vetores selecionados. Na etapa de cruzamento, vetores mutantes são combinados com os vetores originais. Em seguida, ocorre a seleção, onde os novos vetores resultantes do cruzamento são avaliados em relação aos vetores originais. Se um vetor mutante produzir uma solução melhor, ele substitui o vetor original na próxima geração da população. Caso contrário, o vetor original é mantido. O DE segue os passos de mutação, cruzamento e seleção por várias gerações até que um critério de parada seja atingido. O Pseudocódigo 4 descreve a execução completa do DE.

Figura 11. Pseudocódigo do DE

Algoritmo 4: Pseudocódigo do DE

Dados: Tamanho da população (P), Fator de escala (F), Taxa de cruzamento (C_R)

Resultado: Melhor solução encontrada

Inicialize a população aleatoriamente;

enquanto *Critério de parada não for atingido* **faça**

para cada *cada indivíduo i da população* **faça**

 Selecione aleatoriamente três indivíduos diferentes x_1, x_2 e x_3 ;

 Gere o vetor de mutação v_i ;

para cada *cada dimensão j do problema* **faça**

se $\text{rand}(0, 1) < C_R$ **então**

$u_{i,j}$ recebe $v_{i,j}$;

fim

senão

$u_{i,j}$ recebe $x_{i,j}$;

fim

se $\text{fitness}(x_i) < \text{fitness}(u_i)$ **então**

 Substitua x_i por u_i na população;

fim

fim

fim

fim

Fonte: Os autores

O critério de parada do GA, PSO e DE determina quando o processo de otimização deve ser interrompido. Existem várias condições de parada que podem ser consideradas, dependendo das necessidades específicas do problema e dos recursos disponíveis. Algumas das condições de parada mais comuns incluem o número máximo de gerações e convergência da população, onde o algoritmo monitora a diversidade da população ao longo das gerações. Se a diversidade cair abaixo de um certo limiar ou se a melhoria na aptidão estagnar por um número especificado de gerações consecutivas, o algoritmo para.

2.3 Avaliação de Desempenho dos de ML

Para avaliação de desempenho dos modelos preditores foram utilizadas as seguintes métricas: Erro Médio Absoluto (do inglês *Mean Absolute Error* - MAE), Pontuação de Variância Explicada (do inglês *Explained Variance Score* - EVS) e Erro percentual médio absoluto (do inglês *Mean Absolute Percentual Error* - MAPE). A métrica MAE é a diferença média absoluta entre os valores reais e os valores previstos (Equação 2.35). EVS explica a dispersão dos erros (Equação 2.36) e a métrica MAPE mostra a porcentagem de erro em relação aos valores reais (Equação 2.37).

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2.35)$$

$$EVS = 1 - \frac{Var(y_i - \hat{y}_i)}{Var(y)} \quad (2.36)$$

$$MAPE = \frac{1}{n} \sum_{i=1}^n \frac{|y_i - \bar{y}_i|}{\hat{y}_i} \quad (2.37)$$

onde n é o número de amostras, y é o valor verdadeiro, \hat{y} é o valor predito e \bar{y} é a média dos valores verdadeiros.

Analisa-se, também, os resíduos dos modelos propostos para compreensão do viés do modelo para subestimação ou superestimação a partir do cálculo do erro, conforme a Equação 2.38. Os conceitos para subestimação ou superestimação são fundamentais para a interpretação dos resultados e seguem descritos abaixo:

$$e_i = y_i - \hat{y}_i \quad (2.38)$$

- **Modelo com viés de superestimação** - Se os valores de e_i são maiores que zero ($e_i > 0$), significa que os valores previstos \hat{y}_i são persistentemente maiores do que os valores reais y_i . Isso indica que o modelo tende a prever valores mais altos do que os valores reais, resultando em uma tendência de superestimar os valores reais na saída do modelo.
- **Modelo com viés de subestimação** - Por outro lado, se os valores de e_i são menores que zero ($e_i < 0$), o modelo está prevendo persistentemente que \hat{y}_i é menor que y_i . Isso indica que o modelo tende a prever valores mais baixos do que os valores reais, resultando em uma tendência de subestimar os valores reais na saída do modelo.

Esse tipo de análise fornece insights importantes sobre a performance do modelo e ajuda a identificar padrões de sub/superestimação, auxiliando na tomada de decisões

relacionadas ao modelo e aos dados utilizados. Bons modelos de predição equilibram superestimação e subestimação, com distribuições de dados de resíduos e_i simétricas em torno de zero residual, com um número aproximadamente semelhante de amostras antes e depois de zero o que indica que o modelo tem uma capacidade equilibrada de prever valores acima e abaixo dos valores reais.

3 MATERIAL E MÉTODOS

As especificações dos modelos sugeridos para o desenvolvimento da estrutura planejada nesta pesquisa, os dados e métricas utilizados estão descritos a seguir.

3.1 Metodologia Proposta

Propõe-se uma arquitetura de três passos com base em Oliveira *et al.* (2023) para prever o consumo energético e maximizar o ciclo de produção em um instante de tempo $t+1$, utilizando técnicas de modelagem, ML e algoritmos meta-heurísticos. A abordagem deste trabalho propõe a avaliação de múltiplos modelos em cada etapa, selecionando aquele com o melhor desempenho para compor a arquitetura final.

3.1.1 Step 1 - Predição do consumo energético

Este passo envolve a previsão do consumo de energia com base em dados históricos utilizando técnicas de análise temporal. Esta etapa propõe a avaliação de diversos modelos. O objetivo é selecionar o modelo que alcançar uma previsão precisa do consumo energético para o próximo ciclo de produção.

3.1.2 Step 2 - Predição do ciclo de produção

Nesta etapa, são explorados modelos para prever o ciclo de produção a partir da mesma base de dados utilizada no step 1.

3.1.3 Step 3 - Otimização utilizando algoritmos evolutivos

No último passo, são aplicados algoritmos evolutivos para maximizar o ciclo de produção com base no consumo energético previsto pelo melhor modelo do passo 1. Os algoritmos utilizam como função de avaliação o modelo selecionado no passo 2. O espaço de busca é definido pelo vetor previsto no passo 1, com limites superiores e inferiores determinados em função das estratégias de negócio. A solução final é escolhida com base na melhor aptidão encontrada pelos algoritmos após um número específico de gerações e configurações experimentais.

O pré-processamento dos dados e os três passos propostos são apresentados na Figura 12.

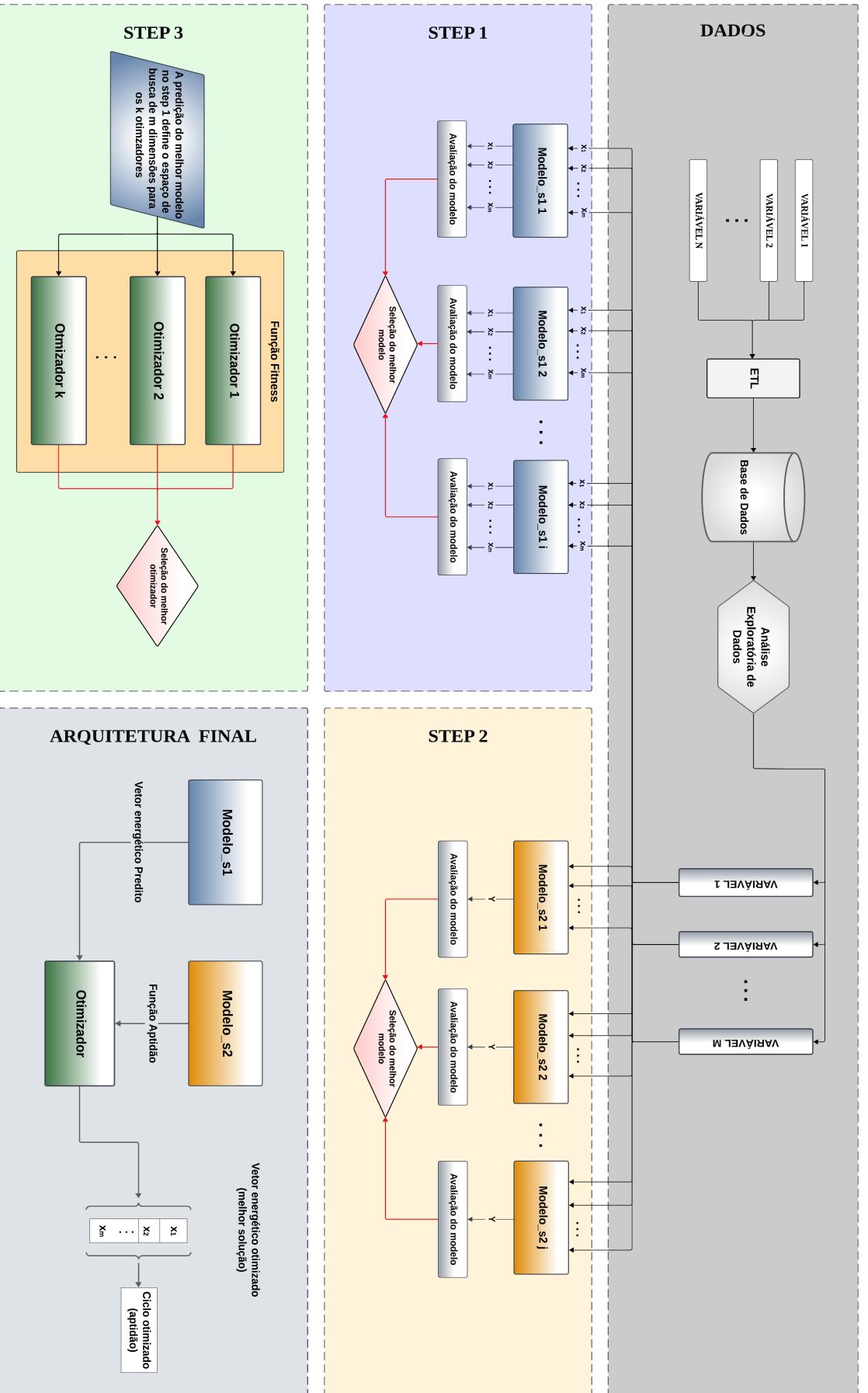


Figura 12. Arquitetura Final

Fonte: os autores

3.2 Banco de Dados: ETL, Análise Exploratória e Pré-processamento de dados

Nesta seção, alguns valores foram suprimidos ou renomeados para preservar a confidencialidade das informações do local da pesquisa e garantir a segurança e integridade dos dados utilizados neste trabalho. Além disso, são apresentadas, no Quadro 2, descrição da máquina utilizada as principais ferramentas para etapas como ETL e visualização de dados.

O *dataset* utilizado nesta pesquisa é o mesmo empregado no estudo realizado por Oliveira et al. (2023), e suas características estão descritas no Quadro 1.

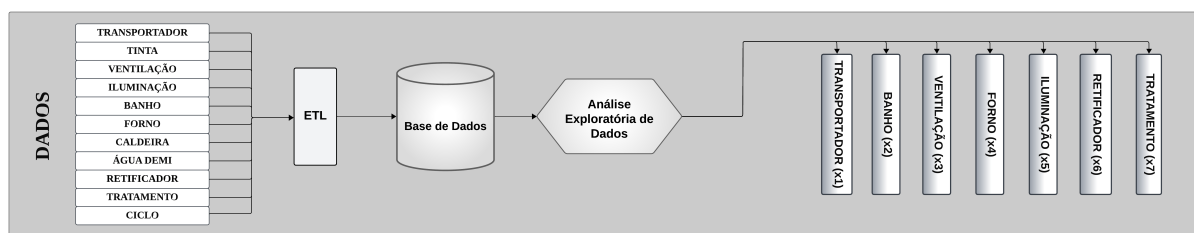
Quadro 1. Dicionário e descrição dos dados

Campo	Tipo	Tamanho	Valores
Data	Timestamp	datetime64	a cada 15 min
Tipo Energia	String	Object	EE e GAS
Máquinas	String	Object	cod máq
Consumo	Float	Float64	KWh
Ciclos	Int	Float64	0 a 15

Fonte: Oliveira et al. (2023)

As amostras foram registradas em intervalos de 15 minutos. Esse intervalo de tempo foi escolhido, em decisão conjunta com o stakeholder, para capturar a dinâmica dos processos de maneira mais precisa e detalhada, permitindo uma análise mais aprofundada dos dados. Os tipos de energia utilizados neste trabalho foram Energia Elétrica(EE) e GAS, que são os maiores ofensores energéticos na planta industrial.

Figura 13. Fluxograma ETL, Análise Exploratória e Pré-processamento de dados.



Fonte: Os autores

Cada máquina na oficina de pintura possui medidores que registram as potências consumidas de EE e GAS associadas ao código da máquina e ao momento da coleta(Data), os valores de Consumo são do tipo *Float* e para ciclos do tipo *int*, conforme Quadro 1. O Fluxograma de tratamento e preparação de dados para a entrada nos modelos implementados nesta pesquisa é apresentado na Figura 13 e estão descritos nas seguintes subseções

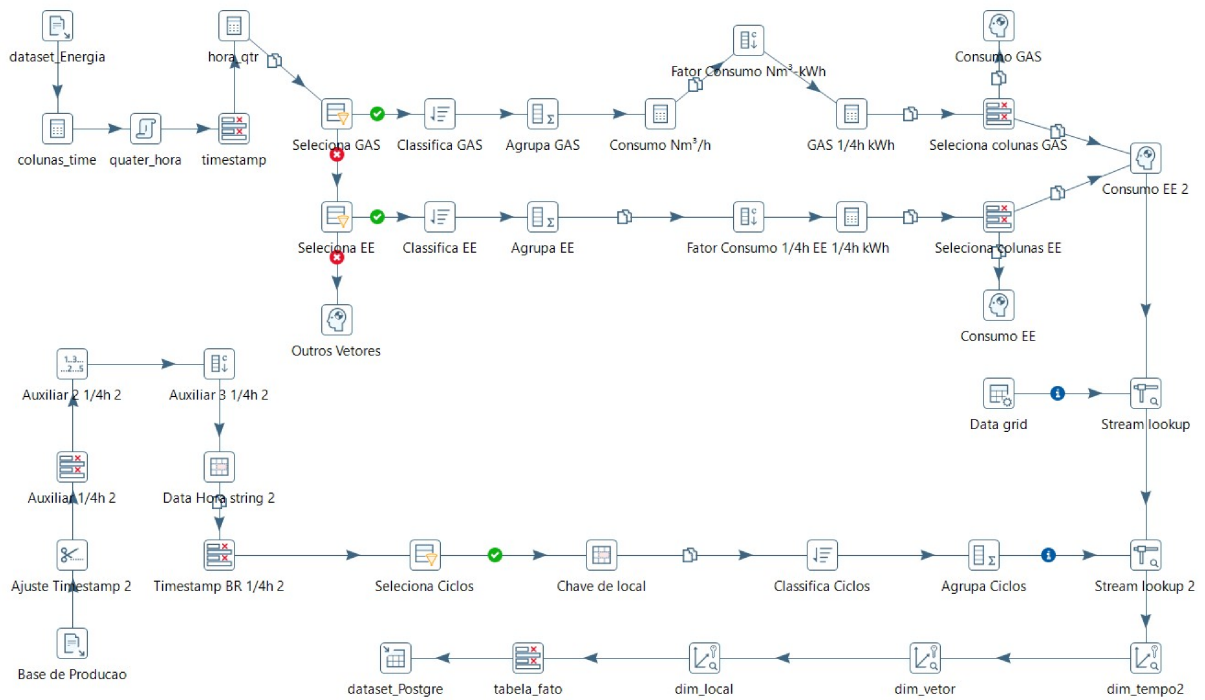
Quadro 2. Especificações de *hardware* e *software* utilizados no projeto.

Categoria	Componente	Descrição
Máquina	Processador	AMD Ryzen 5, 8 núcleos, 20 MB de <i>cache</i> , 4.70 GHz
	Placa de vídeo	NVIDIA GeForce RTX 3050, 4 GB GDDR6
	Memória RAM	16 GB DDR5, 4800 MHz
	Armazenamento	1 TB SSD
	Sistema Operacional	Windows 11
Linguagens de Programação	Python	Linguagem interpretada, de alto nível e multiparadigma.
	<i>Structured Query Language (SQL)</i>	Linguagem de consulta estruturada usada para manipular e gerenciar bancos de dados relacionais.
ETL	Pentaho	Plataforma <i>open-source</i> para ETL.
	PostgreSQL	Sistema de gerenciamento de banco de dados relacional <i>open-source</i> .
Manipulação de Dados	Pandas	Biblioteca Python para manipulação e análise de dados.
	NumPy	Biblioteca Python para computação numérica.
Visualização de Dados	Matplotlib	Biblioteca Python para criação de gráficos.
	Seaborn	Biblioteca Python baseada no Matplotlib para visualização estatística de dados.
ML e DL	TensorFlow	Plataforma open-source para ML e DL.
	Scikit-learn	Biblioteca Python para ML e mineração de dados.

3.2.1 Extração, Transformação e Carga

A técnica de ETL envolve o processo de coleta (extração), limpeza e transformação e carregamento de dados das bases de dados relacionadas ao consumo de energia (Kwh) e à produção de veículos (ciclos), conforme Figura 14.

Figura 14. Fluxograma - Etapa ETL



Fonte: Os autores

O ETL foi realizado utilizando a ferramenta Pentaho, realizando o tratamento e carga dos dados na base de dados PostgreSQL. Ambas as ferramentas estão melhor descritas no Quadro 2.

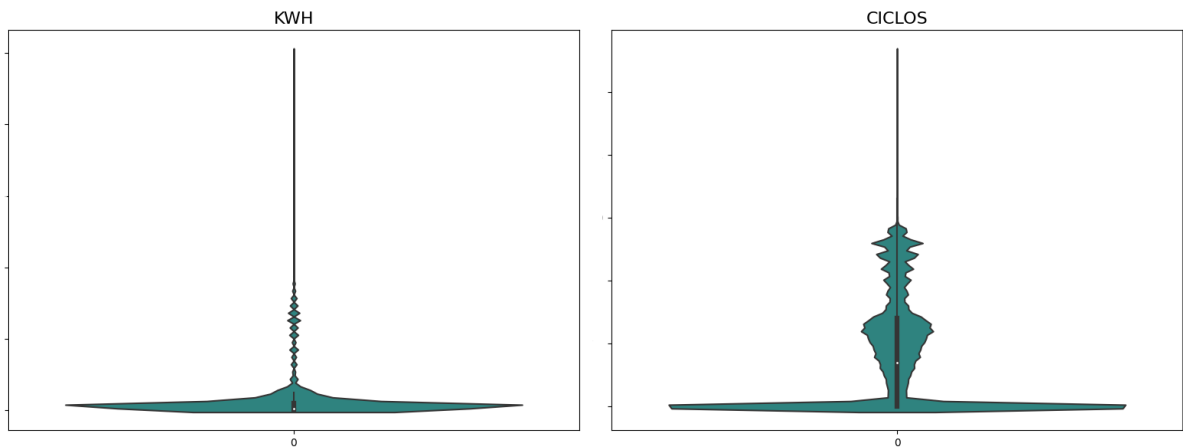
Após a inserção das bases de dados "`dataset_Energia`" e "`Base de Produção`" no fluxograma de ETL, os dados foram tratados para que o `dataset` final fosse construído com uma estruturação conforme as especificações descritas no Quadro 1.

O processo de ETL foi essencial para garantir a qualidade e a integridade dos dados utilizados nesta pesquisa. As distribuições destas duas bases de dados após esta etapa estão descritas na Figura 15.

3.2.2 Período dos Dados Coletados

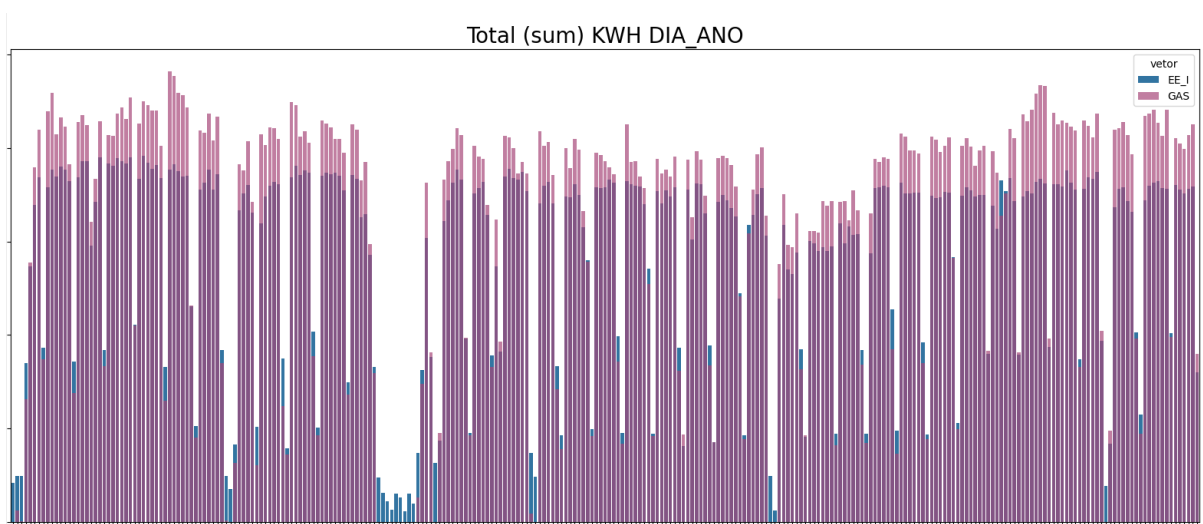
O período dos dados coletados para treinamento e teste abrange de janeiro a setembro de 2023. Esse período foi selecionado com base na disponibilidade dos dados e na relevância temporal para a análise proposta.

Figura 15. Violinplot Consumo (Kwh) & Ciclos.



Fonte: Os autores

Figura 16. Consumo diário de Kwh de janeiro a setembro de 2023.



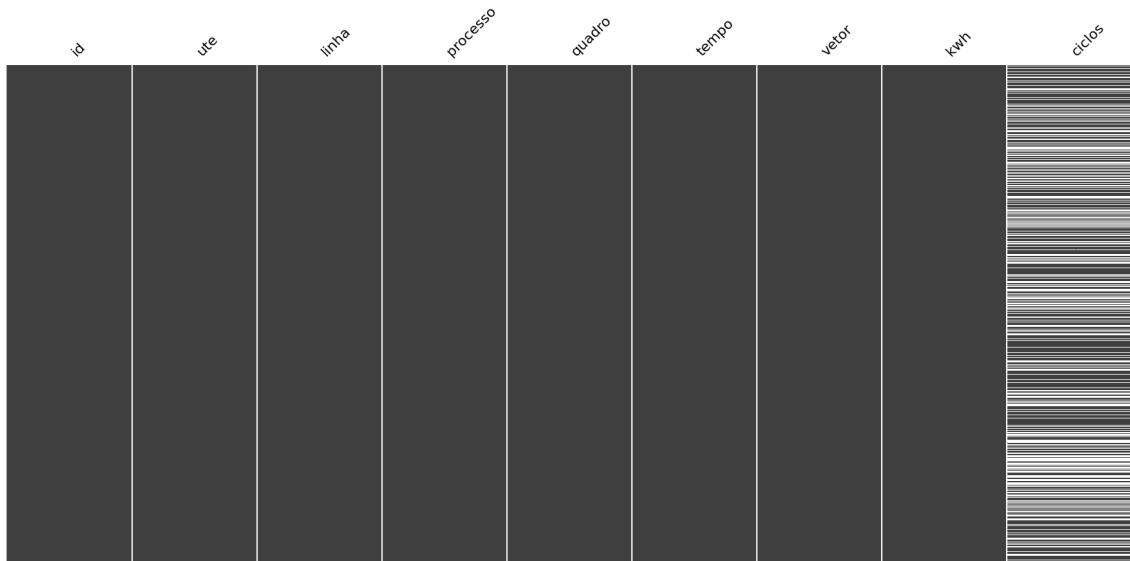
Fonte: Os autores

Os valores diários para EE e GAS estão descritos na Figura 16 e apresentam grande variabilidade nos dados, não sendo possível identificar visualmente um padrão de comportamento no consumo tanto para EE quanto para GAS.

3.2.3 Tratamento de Valores Ausentes

Os valores ausentes na variável Ciclos, que totalizam 1338425 amostras, foram interpretados e substituídos por zero, conforme orientação do stakeholder responsável. Esse tipo de tratamento é fundamentado em trabalhos e podem garantir, quando justificáveis, a consistência e a completude dos conjuntos de dados utilizados.

Figura 17. Dados null para a variável ciclos.



Fonte: Os autores

3.2.4 Os outliers

Ao analisar as variáveis a partir do gráfico de dispersão e de distribuição violinplot, conforme Figura 18, verificou-se a Presença de outliers nas observações das variáveis (processos) do *dataset*. Esses pontos fora do padrão podem ter surgido por diversos motivos, como erros de medição, eventos raros ou situações incomuns e tem a capacidade de influenciar significativamente a performance dos modelos de ML e DL. Como estratégia, optou-se pela busca de modelos que apresentassem maior robustez e que fossem menos sensíveis a *outliers*.

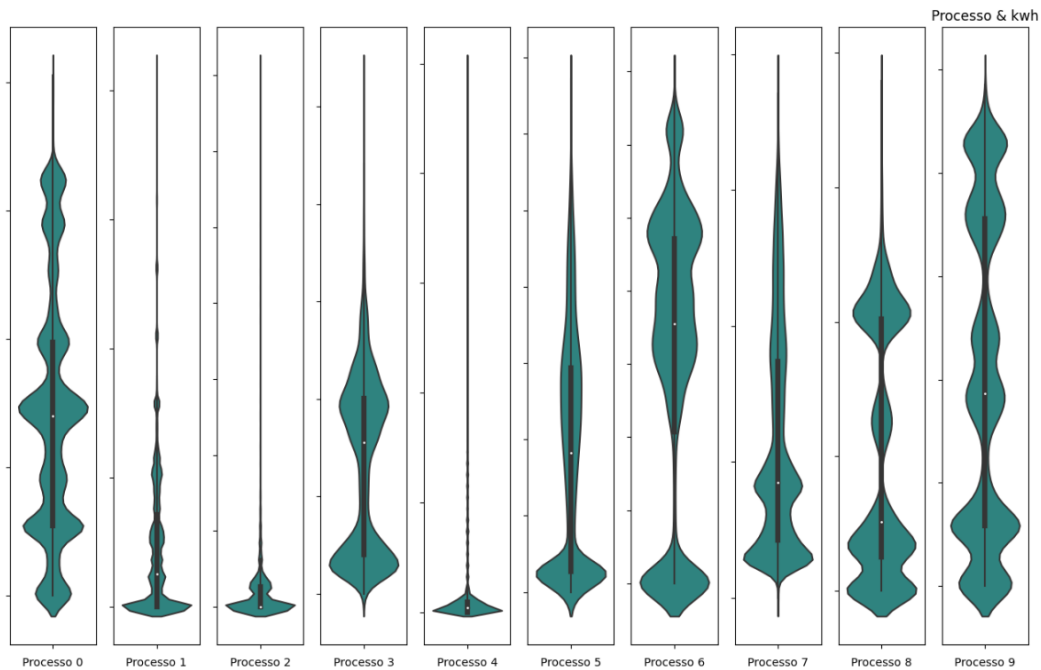
3.2.5 Seleção de variáveis

A seleção de variáveis é uma etapa crítica no processo de análise de dados, pois visa identificar e manter apenas as variáveis mais relevantes e significativas para o objetivo da análise. Nesse contexto, após a conclusão da EDA, representada na Figura 1, optou-se por remover três variáveis do banco de dados. Essa decisão foi embasada na identificação das variáveis que apresentavam menor correlação, utilizando a métrica de correlação de Pearson.

$$r_{xy} = \frac{n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{\sqrt{(n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2)(n \sum_{i=1}^n y_i^2 - (\sum_{i=1}^n y_i)^2)}} \quad (3.1)$$

Onde, r_{xy} é o coeficiente de correlação de Pearson entre as variáveis x e y ; n é o número de observações; x_i e y_i são os valores das variáveis x e y para cada observação

Figura 18. Violinplot consumo (Kwh) por processos de produção.



Fonte: Oliveira et al. (2023)

i. $\sum_{i=1}^n x_i$ e $\sum_{i=1}^n y_i$ são as soma dos valores de x e y para todas as observações; $\sum_{i=1}^n$ e $\sum_{i=1}^n x_i^2$ são as somas dos quadrados dos valores de x e y para todas as observações.

Conforme os cálculos da Correlação de Pearson apresentados na Figura 19, as variáveis caldeira, água demi e tinta foram identificadas como aquelas com os valores de menor correlação em relação às demais variáveis do conjunto de dados e foram removidas do conjunto de dados final utilizado nas análises subsequentes deste estudo.

É relevante destacar que todas as análises realizadas neste estudo foram conduzidas utilizando o conjunto de dados resultante da remoção das três variáveis mencionadas. Esse novo conjunto de dados consiste em sete variáveis relacionadas ao consumo de energia em kWh, juntamente com uma variável representando o número de ciclos.

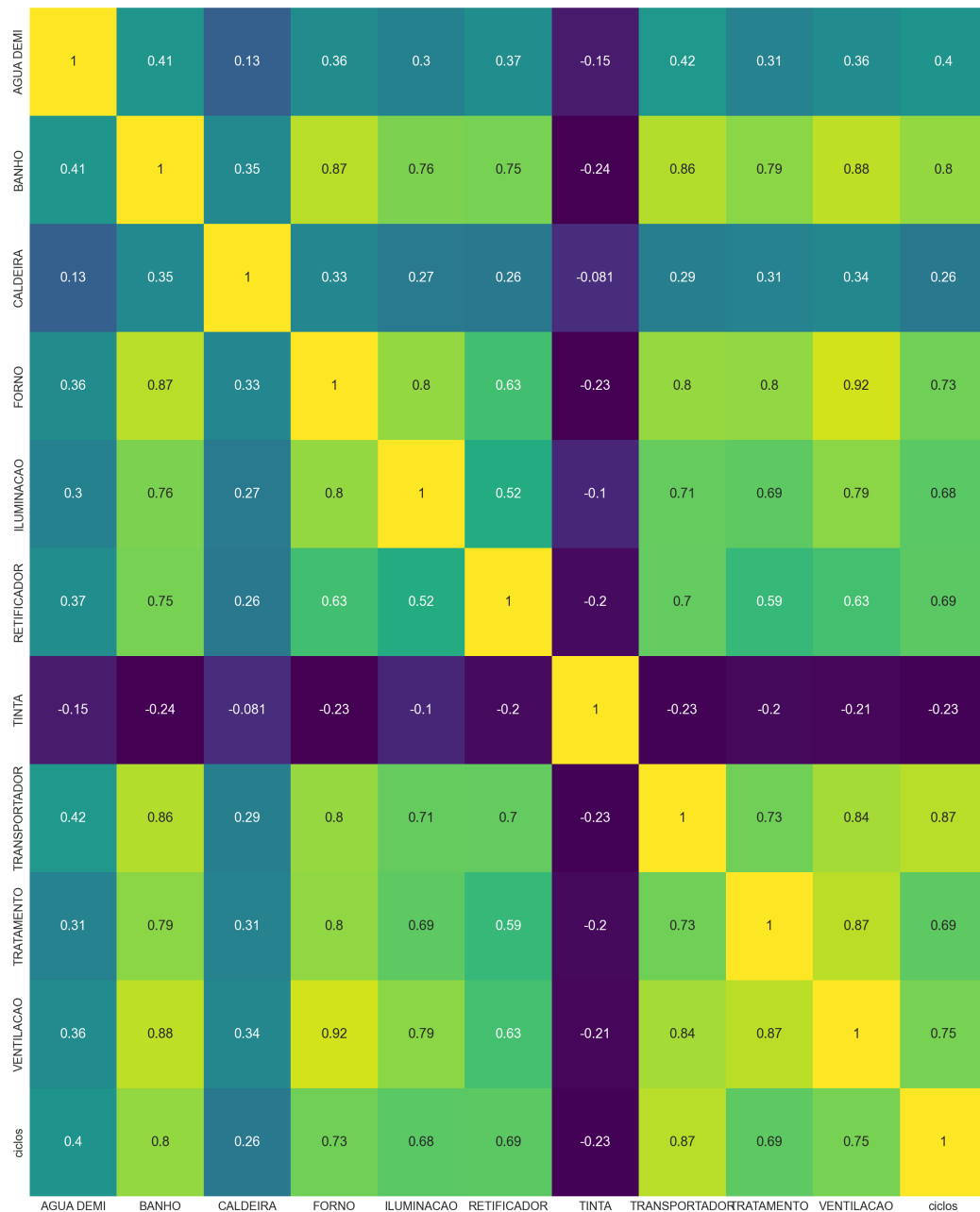
3.3 Desenvolvimento da Arquitetura Proposta

Esta seção apresenta os modelos, suas configurações e variáveis de entrada experimentados neste trabalho para cada *step* apresentado na arquitetura da seção 3.1.

3.3.1 Step 1 - Predição do consumo energético

Além dos modelos implementados por Oliveira *et al.* (2023), utilizou-se também o modelo híbrido GRU-LSTM como nova proposta de implementação. A arquitetura do modelo GRU-LSTM conta com uma camada GRU, uma camada LSTM e duas camadas

Figura 19. Correlação de Pearson.

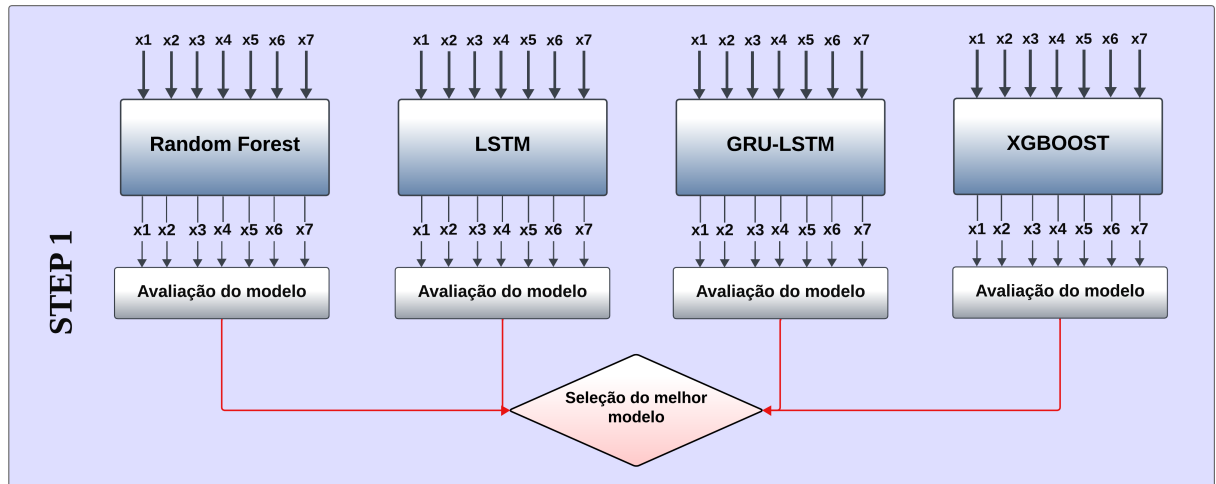


Fonte: Os autores

Densas. Os modelos são treinados com os dados do consumo energético de cada processo e têm como target $t + 1$, portanto, o objetivo é estabelecer uma previsão precisa do consumo energético para o próximo ciclo de produção. O *step* 1 pode ser observado na Figura 20.

Cada amostra do dataset é um vetor de sete valores de $KWh(X)$ correspondentes aos sete processos do setor de pintura em um dado tempo t , os quais são as variáveis utilizadas neste estudo. Tanto a GRU como a LSTM têm como entrada um conjunto de n amostras do dataset, onde n é o valor do lag. Assim, se desejamos utilizar 3 amostras ($\text{lag}=3$) para prever a quarta, temos como entrada no modelo $[X_{t-2}, X_{t-1}, X_t]$ e $[X_{t+1}]$

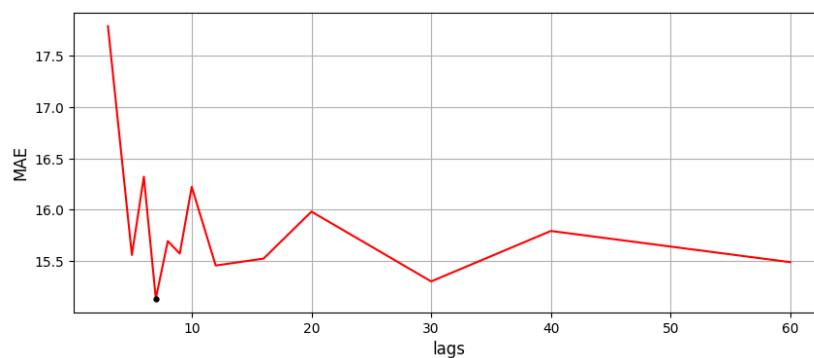
Figura 20. *Step 1* da arquitetura proposta.



Fonte: os autores

como saída. O valor do lag foi otimizado em função do modelo híbrido GRU-LSTM com auxílio da função `ParameterGrid` da biblioteca `scikit-learn` a fim de buscar valores para o lag que pudessem gerar os melhores resultados para o modelo híbrido implementado neste *step 1* da arquitetura proposta. Na Figura 21 são apresentados os MAEs obtidos para cada lag em [3, 5, 6, 7, 8, 9, 10, 12, 16, 20, 30, 40, 60]. O lag igual a 7 apresentou o melhor valor de MAE para o modelo híbrido, portanto, são 49 valores de KWh distribuídos em 7 vetores utilizados como entrada tanto na GRU como na LSTM.

Figura 21. Variação do erro em função do número de *steps*.



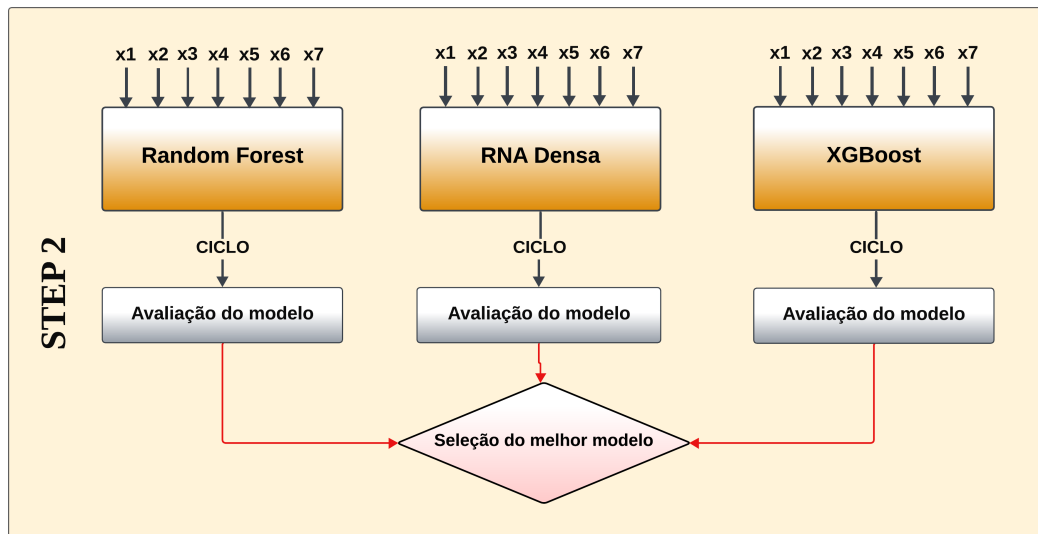
Fonte: os autores

Os hiperparâmetros definidos para os modelos para deste *step* estão descritos na Tabela 1.

3.3.2 Step 2 - Predição Do Ciclo De Produção

No segundo passo, segue-se a arquitetura proposta para os modelos RF, RNA Densa e XGBoost. Utilizou-se a mesma base de dados do *step 1* deste trabalho para treinamento dos modelos que preveem o ciclo de produção de uma pintura automotiva, conforme observado na Figura 22. Os hiperparâmetros de cada modelo estão descritos na Tabela 1.

Figura 22. *Step 2* da arquitetura proposta.



Fonte: os autores

Tabela 1. Hiperparâmetros dos modelos e otimizadores.

Modelo	Principais Parâmetros
RF	estimators=50, min_samples_leaf=6
XGBoost	max_depth=6, learning_rate=0,3
GRU-LSTM	Units=32, loss=mse, optimizer=adam, epoch=400, batch_size=16
RNA Densa	activation (hidden layers)=sigmoid, activation (output layer)=relu
GA	mutation_probability=0,01, elitism=0,1, crossover_probability=0,7
DE	crossover_probability=0.7, F=0,6
PSO	w=0,1; c ₁ =0.1; c ₂ =0,2

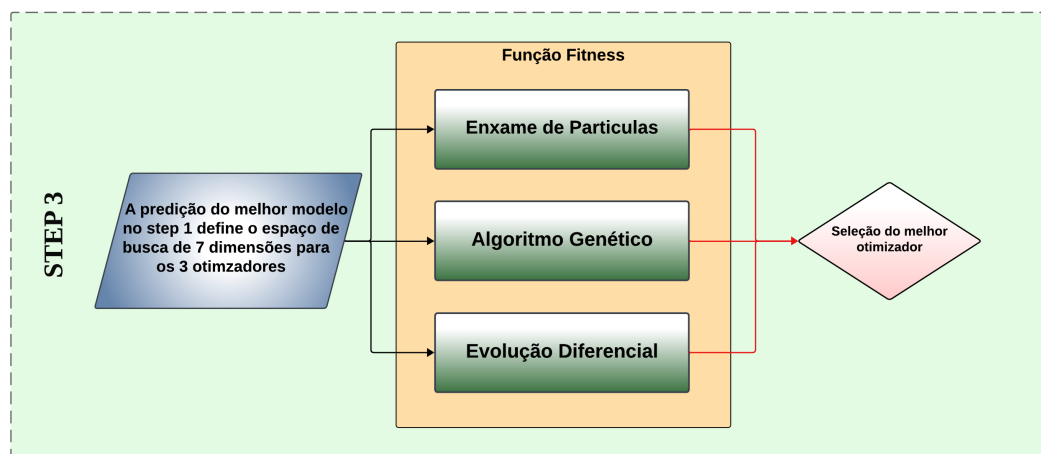
Fonte: Os autores

3.3.3 Step 3 - Otimização Utilizando Algoritmos Evolutivos

No último passo, são empregados algoritmos evolutivos GA, PSO e DE conforme observado na Figura 23. Os limites superiores do espaço de busca são definidos pelo

vetor predito no passo 1, enquanto os limites inferiores são definidos por uma redução percentual (definida pelas estratégias de negócio) do mesmo vetor. Portanto, cada indivíduo da população inicial dos otimizadores representa um vetor de consumo energético e uma possível solução do problema, cuja aptidão é o número de ciclos de produção. O otimizador que, ao final de n gerações, apresentar a melhor solução será escolhido para compor o terceiro passo da arquitetura. Os três otimizadores foram definidos com uma população de 50 indivíduos e otimizados ao longo de 100 gerações. Outros parâmetros, específicos de cada otimizador, foram definidos a partir de diferentes experimentações e são apresentados na Tabela 1.

Figura 23. *Step 3* da arquitetura proposta.



Fonte: Os autores

4 Resultados e Discussão

Nesta seção são apresentados os resultados dos modelos e otimizadores de cada *step*, discussão dos resultados e arquitetura final de 3 *steps*.

4.1 Resultados do *Step* 1

Neste *step*, os 4 modelos foram avaliados a partir das métricas MAPE e EVS . Optou-se por utilizar o MAPE para melhor compreensão do erro dos valores preditos de consumo energético em $t+1$, pois essa métrica expressa de forma direta o quanto as previsões se desviam dos valores reais em termos percentuais.

Tabela 2. Avaliação dos modelos no *step* 1

Modelo	MAPE	EVS
RF	10,1%	0,85
LSTM	8,2%	0,91
XGBoost	7,9%	0,88
GRU-LSTM	7,7%	0,908

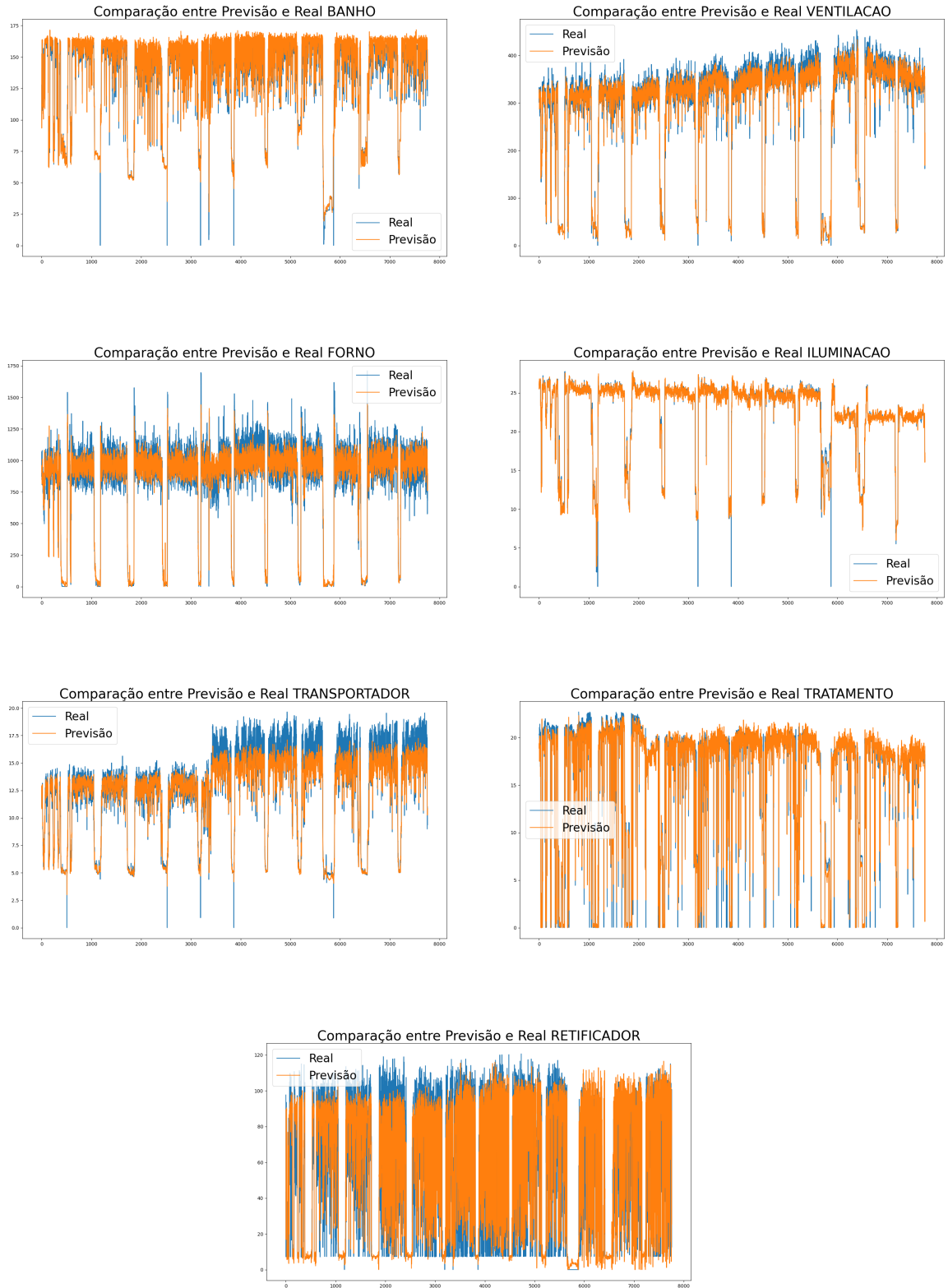
Fonte: Os autores

Entre os modelos avaliados, a LSTM apresentou os melhores valores para o EVS, atingindo 0,91. Por outro lado, a GRU-LSTM obteve o melhor resultado para o MAPE, além de ser o segundo melhor em relação ao EVS, alcançando 0,908. Dessa forma, escolheu-se a GRU-LSTM como o melhor modelo para o *step* 1.

O MAPE da GRU-LSTM é 0,5% menor e o EVS é apenas 0,002 maior quando comparado com a LSTM, conforme detalhado na Tabela 2, evidenciando sua superioridade em termos de eficácia na previsão do consumo energético. Os resultados estão descritos na Tabela 2. Os valores preditos versus os valores reais para cada uma das variáveis utilizando o melhor modelo deste *step* estão descritas na Figura 24.

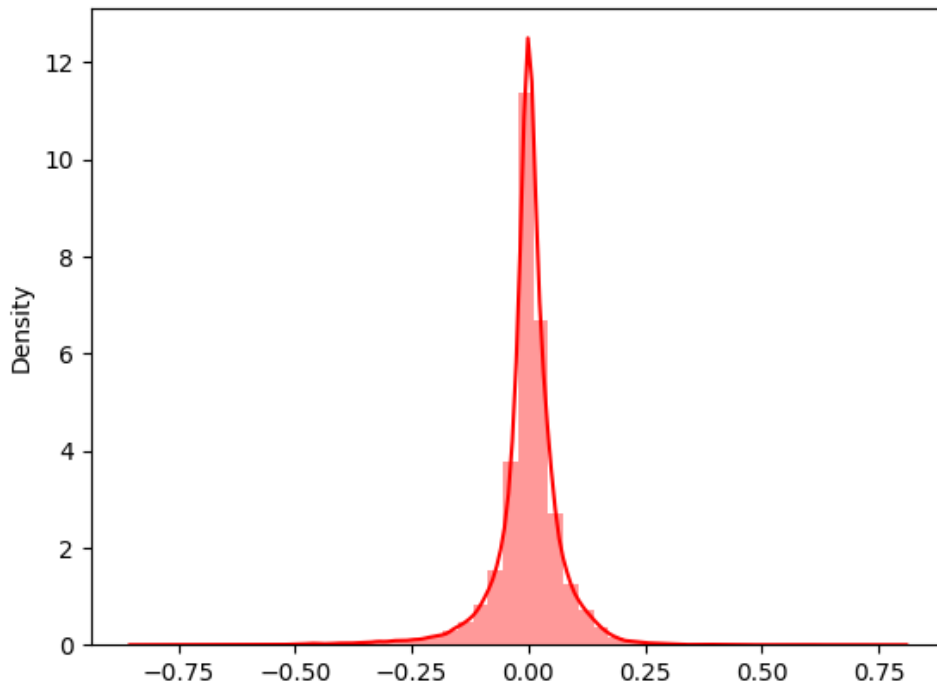
O gráfico residual (Figura 25) apresenta os valores em densidade, tanto para subestimação quanto superestimação. Assim, é possível analisar estes resíduos do melhor modelo usando o gráfico de distribuição de e_i em torno de zero. Um gráfico de distribuição dos resíduos da GRU-LSTM é simétrico, com uma curva de sino, indica que o modelo está distribuindo os erros de forma uniforme, sem viés significativo para superestimação ou subestimação, equilibrado e menos tendencioso. Avalia-se que a qualidade e a confiabilidade das previsões geradas pelo modelo de regressão podem ser efetivas.

Figura 24. Ajuste das variáveis para o modelo híbrido.



Fonte: os autores

Figura 25. Análise residual da GRU-LSTM



Fonte: os autores

4.2 Resultados do *Step 2*

Para a avaliação dos modelos no *step 2*, optou-se pela MAE devido à sua capacidade de proporcionar uma visão direta do erro absoluto dos modelos em relação à quantidade de veículos produzidos. Esta escolha foi feita visando melhorar a análise dos resultados por parte dos stakeholders envolvidos no processo, uma vez que o MAE permite uma interpretação mais intuitiva do desempenho dos modelos em termos de magnitude dos erros cometidos na predição com dados na base de teste.

Ao utilizar o MAE, os stakeholders podem compreender de forma mais clara e objetiva o quanto os modelos se desviam dos valores reais de produção de veículos. Isso é crucial para tomar decisões informadas e para ajustar estratégias e processos com base nas previsões geradas pelos modelos. Além disso, o MAE é uma métrica amplamente reconhecida e de fácil interpretação.

Além do MAE, EVS foi novamente utilizado neste *step*. Esta métrica é significativa para avaliar a capacidade dos modelos em explicar a variação dos dados em relação à quantidade de veículos produzidos. A inclusão do EVS complementa a análise ao fornecer uma medida adicional de desempenho, permitindo uma avaliação mais abrangente da precisão e da capacidade explicativa dos modelos desenvolvidos.

Na Tabela 3, o modelo RNA Densa se destaca como melhor modelo na predição dos ciclos de produção ao alcançar os melhores resultados para MAE (1,56) e EVS (0,85).

Tabela 3. Avaliação dos modelos no *step 2*

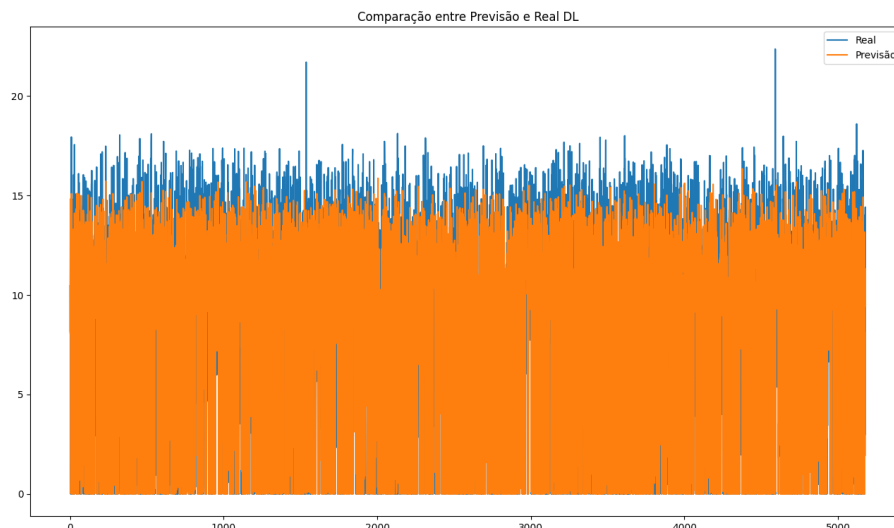
Modelo	MAE	ES
RF	1,62	0,84
RNA Densa	1,56	0,85
XGBoost	1,66	0,84

Fonte: Os autores

Esses resultados indicam que o modelo RNA Densa apresenta um desempenho superior em termos de precisão na predição dos ciclos de produção de veículos, uma vez que o MAE de 1,56 representa um erro absoluto médio relativamente baixo em relação à quantidade real de veículos produzidos, enquanto o EVS de 0,85 mostra que o modelo é capaz de explicar 85% da variação nos dados de produção.

O modelo RNA Densa, a partir dos valores das métricas de avaliação utilizadas neste trabalho, mostra-se efetivo em capturar padrões e tendências nos dados de produção, tornando-se uma ferramenta útil para os stakeholders no processo de planejamento e tomada de decisões estratégicas. Ao proporcionar resultados mais precisos e explicativos, o modelo RNA Densa contribui para otimizar os processos de produção e maximizar a eficiência operacional da indústria.

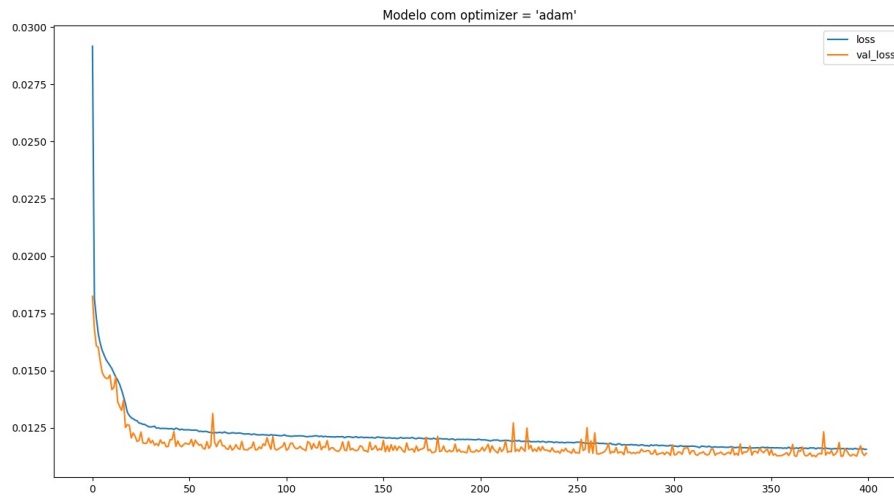
Figura 26. Predição do ciclo de produção utilizando o modelo DL.



Fonte: os autores

As Figuras 26 e 27 complementam a análise ao apresentarem visualmente a predição de ciclos do modelo RNA Densa e a curva de convergência de treino e validação após 400 épocas de treinamento, oferecendo uma compreensão mais abrangente do desempenho e da capacidade do modelo em lidar com os dados de produção em um ambiente de aplicação e destacando, também sua capacidade de generalização.

Figura 27. Curvas de convergência: modelo RNA Densa.



Fonte: os autores

A Figura 26 mostra as previsões geradas pelo modelo RNA Densa para os ciclos de produção, permitindo uma avaliação visual da efetividade do modelo em realizar previsões em relação aos valores reais.

Na Figura 27, apresenta-se a curva de convergência do treinamento e validação do modelo ao longo de 400 épocas, revelando a trajetória do aprendizado do modelo e sua capacidade de generalização para dados não vistos durante o treinamento. Essa análise visual da curva de convergência permite inferir que a estabilidade e a eficiência, do modelo selecionado, são relativamente altas e sugerem que o modelo possui capacidade de aprender com os dados e fazer previsões confiáveis em diferentes contextos.

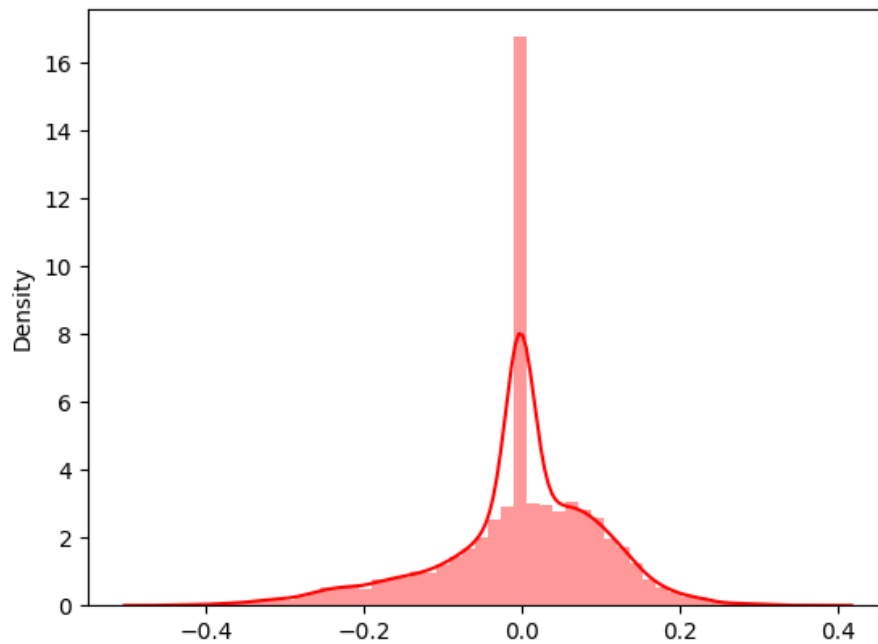
Os valores para a análise de subestimação e superestimação para o *step 2*, estão apresentados na Figura 28, também, em um gráfico de distribuição de e_i em torno de zero. Observa-se que o gráfico de distribuição dos resíduos da RNA Densa não é tão simétrico, mas, possuindo uma curva de sino com um modelo que está distribuindo os erros de forma uniforme, com um pequeno viés para superestimação.

É possível observar que a quantidade de dados de resíduos $e_i > 0$ é ligeiramente maior que a quantidade de dados de $e_i < 0$, o que demonstra que o modelo RNA Densas possui uma pequena tendência para superestimação. Como os dados para superestimação não são tão preponderantes que os de subestimação, é possível inferir que o modelo não é tendencioso para superestimação e subestimação.

4.3 Resultados do *Step 3*

O *step 3* é dependente dos melhores modelos selecionados nos *steps* anteriores. O melhor modelo do *step 1* faz a previsão do vetor de consumo energético com sete variáveis

Figura 28. Análise residual da RNA Densa



Fonte: os autores

que se torna espaço de busca para os otimizadores testados neste *step*, conforme Figura 23. O melhor modelo do *step 2* é utilizado como a função *fitness* para os três otimizadores.

Como fluxo de ações realizadas neste trabalho, ao iniciar a execução das ações do *step 3*, parte da arquitetura já estava definida da seguinte forma: Utilização da GRU-LSTM para o *step 1* e a RNA Densa para o *step 2*.

A previsão gerada pela GRU-LSTM no tempo $t+1$, a partir de uma amostra da base de teste, que o valor de ciclo produtivo igual a 5,58, resultou em um vetor de consumo energético, cujo consumo total é de 1818,0 KWh. Os valores do vetor predito foram estabelecidos como o limite superior do espaço de busca, enquanto o limite inferior foi determinado aplicando uma redução de 20% em relação ao limite superior. Isso significa que todas as possíveis soluções investigadas por cada otimizador abrangem um intervalo de busca que representa de 80% a 100% do consumo máximo previsto no *step 1*. A redução de 20% é o limiar que foi definido de forma arbitrária, a partir de discussões e alinhamento com *stakeholders* do processo.

A Tabela 4 apresenta de forma detalhada os resultados obtidos pela melhor solução de cada otimizador após 100 gerações. O algoritmo DE se destacou ao alcançar o melhor desempenho na redução do consumo de energia, apresentando o melhor resultado em otimização do vetor energético de 1818,0 KWh (*step 1*) para 1517,15 kwh, conseguindo uma diminuição de até 300,80 KWh. Em relação ao ciclo produtivo, o algoritmo PSO se sobressaiu, maximizando o valor do ciclo predito no *step 2* (5,58 ciclos) para 12,70 ciclos.

Notavelmente, o PSO possibilita uma maximização de até 127% em comparação com o valor de ciclo produtivo, enquanto também alcança uma redução de 8,2% no consumo energético.

Tabela 4. Desempenho dos otimizadores no *step 3*

Otimizador	Total (kWh)	Redução (kWh)	Ciclo (<i>fitness</i>)
GA	1527,01	290,99	10,58
DE	1517,15	300,80	10,66
PSO	1668,88	149,12	12,70

Fonte: Os autores

Os resultados para o algoritmo DE demonstram uma maximização significativa de 91% para o ciclo produtivo e uma redução de 16,5% no consumo energético. Apesar de apresentar um valor de ciclo 36% menor que o PSO, a redução do consumo energético alcançada pelo DE é duas vezes maior que a obtida pelo PSO. Diante desses resultados, o algoritmo DE foi escolhido para integrar a arquitetura final no *step 3*.

Os valores encontrados ressaltam a importância de selecionar o algoritmo de otimização mais adequado para cada objetivo específico, pois cada um apresenta vantagens e desvantagens, influenciando diretamente nos resultados obtidos e nas melhorias efetivas nos processos produtivos. A integração do DE na arquitetura final do *step 3* representa uma decisão estratégica, a partir de alinhamentos com o stakeholder, baseada na busca por eficiência energética.

Tabela 5. Combinação de todos os modelos do *step 2* e *step 3*.

Step 2	Step 3	Total (kWh)	Redução (kWh)	Ciclo (<i>fitness</i>)
RF	GA	1651,4886	166,518	9,7276
RF	DE	1653,45529	164,5519	9,888
RF	PSO	1698,6149	119,3922	9,57699
RNA_Densa	GA	1527,0123	290,9809	10,5803
RNA_Densa	DE	1517,1503	300,7071	10,6603
RNA_Densa	PSO	1668,8798	149,1225	12,6903
XGB	GA	1688,7793	129,2279	11,3403
XGB	DE	1691,8287	126,1785	12,437
XGB	PSO	1751,8158	66,1913	11,3847

Fonte: Os autores

Na tabela 5 são apresentados os resultados das outras possíveis combinações dos modelos considerados neste trabalho. Dado que a escolha do melhor modelo do *step 1* é extrínseca aos outros *steps*, a escolha foi baseada no desempenho individual de cada modelo. Desta forma, os modelos do *step 1*, responsáveis por definirem o espaço de busca no *step 3*, não foram incluídos na combinação de modelos. O espaço de busca do *step 3* foi

fixado com o vetor predito pelo melhor modelo do *step* 1, cujo consumo total foi 1818,0 KWh, proporcionando uma avaliação mais justa dos resultados finais. Por outro lado, o *step* 2 é intrínseco ao *step* 3, um vez que é responsável por definir a função *fitness* dos otimizadores, impactando diretamente nos resultados.

4.4 Arquitetura Final e discussões

A arquitetura final é composta pelos modelos GRU-LSTM, RNA Densa e DE nos *steps* 1, 2 e 3, respectivamente, conforme o fluxograma de execução das atividades realizadas e sua consolidação proposta como objetivo para o presente estudo, observado na Figura 29. Essa estrutura é resultado de um processo analítico e de modelagens em inteligência artificial, nos quais cada etapa desempenha um papel relevante na obtenção de resultados. No *step* 1, o modelo GRU-LSTM foi selecionado por sua capacidade de lidar os dados de séries temporais que compõem a base de dados analisados neste estudo.

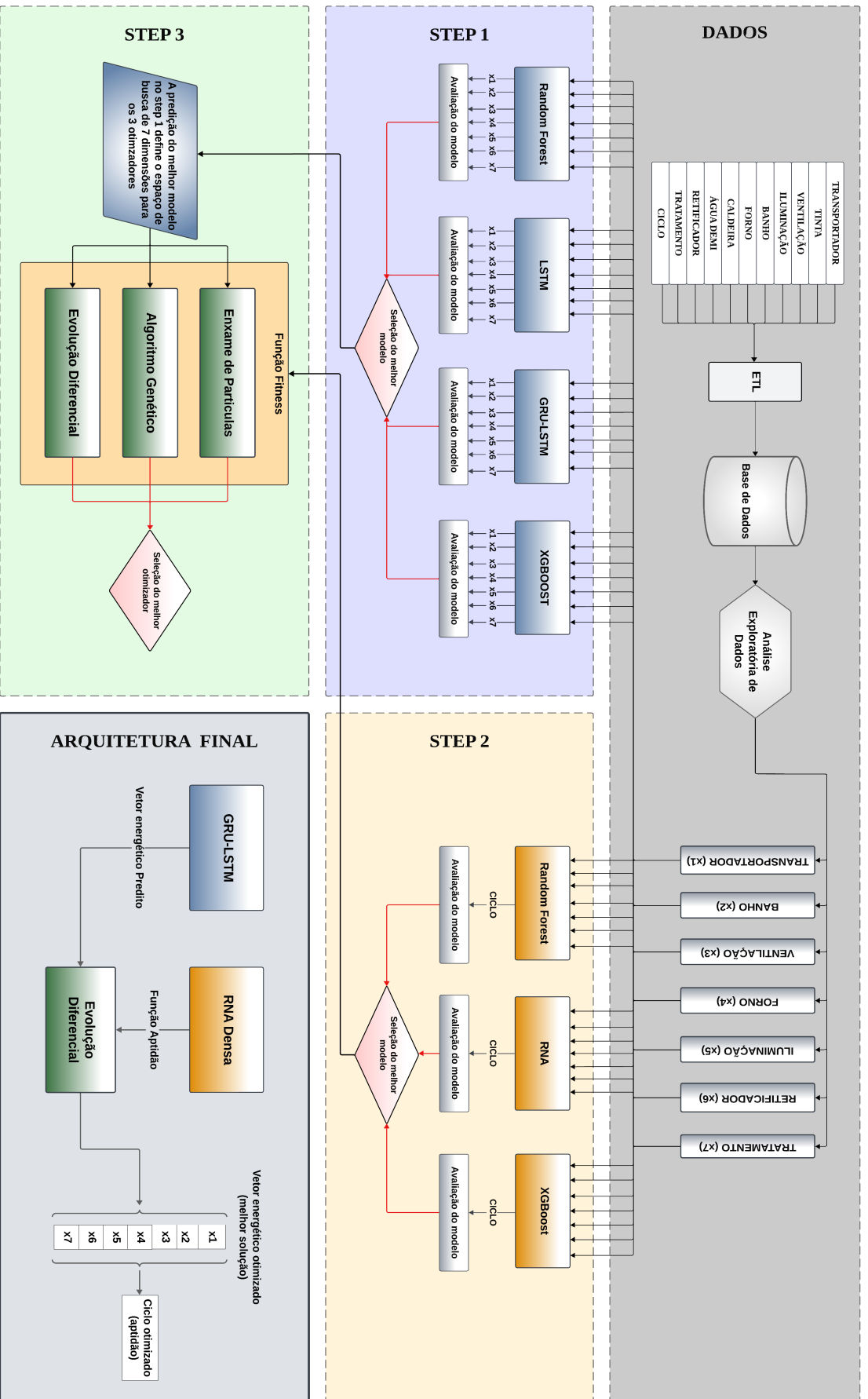
No *step* 2, a RNA Densa apresentou melhores resultados devido à sua capacidade de generalização e desempenho em termos de precisão na predição dos ciclos de produção de veículos, demonstrando ser uma ferramenta eficaz para capturar tendências e padrões nos dados de produção, contribuindo para as predições nos processos industriais. Para o *step* 3, o algoritmo de otimização DE foi escolhido devido aos seus resultados promissores na redução do consumo energético e na maximização da eficiência operacional.

Assim, a Figura 29 representa de forma visual a integração e a interdependência dos modelos nos diferentes *steps*, destacando a contribuição de cada um na consecução dos objetivos deste trabalho científico e na melhoria dos processos industriais abordados. A arquitetura final não só representa uma solução técnica robusta, mas também evidencia o potencial das metodologias e técnicas empregadas para impulsionar a inovação e a eficiência na indústria.

Os resultados apresentados neste estudo refletem não apenas a aplicação dos métodos convencionais de experimentação, mas também a integração das estratégias de negócios definidas pelos stakeholders participantes do processo. Essas estratégias abrangem desde o processamento inicial dos dados até a etapa final da arquitetura de passos adotada. Os resultados obtidos demonstram melhorias significativas em comparação com trabalhos anteriores (OLIVEIRA et al., 2023), alcançadas não apenas pela introdução de novos modelos e otimizadores, mas também pela revisão da análise de dados na fase de pré-processamento. Especificamente, a remoção das variáveis com baixa correlação de Pearson representou uma mudança crucial que contribuiu para o aprimoramento dos resultados obtidos.

É importante salientar que, os resultados obtidos neste trabalho estão alinhados com as descobertas de estudos realizados pela academia. Autores como Hong et al. (2024),

Figura 29. Arquitetura Final



Fonte: os autores

Zafar et al. (2022) e Kervanci, Akay e Özceylan (2024), têm apresentado trabalhos que destacam modelos híbridos, como o GRU-LSTM, em diferentes contextos, evidenciando sua eficácia em comparação com outros modelos analisados. O sucesso desses modelos híbridos é atribuído à sua capacidade de capturar relações temporais complexas e de longo prazo de maneira eficiente.

Embora modelos como RF e XGBoost não tenham uma arquitetura específica projetada para capturar explicitamente relações temporais, eles ainda conseguem obter resultados semelhantes aos modelos GRU-LSTM e RNA Densas. Isso se deve às suas arquiteturas ensemble e à sua flexibilidade em lidar com uma ampla variedade de padrões de dados.

5 Considerações Finais e Trabalhos Futuros

A nova abordagem composta pela arquitetura de 3 *steps* mostrou-se altamente promissora na previsão e otimização do consumo energético em oficinas de pintura automotiva. A arquitetura final foi capaz de realizar previsões precisas e reduzir o consumo energético em até 16,5%, ao mesmo tempo em que maximizou o ciclo produtivo. Isso foi alcançado por meio da combinação dos modelos GRU-LSTM e RNA Densas para previsão, juntamente com o uso do algoritmo meta-heurístico DE. Os resultados obtidos demonstraram o potencial significativo dessa abordagem integrada de técnicas de ML, DL, abordagens híbridas e algoritmos evolutivos.

Propõe-se, como trabalhos futuros, uma discussão para a expansão dessa solução para outras unidades de Pintura similares e , ainda, para as oficinas de Funilaria e Montagem, ou qualquer outro processo desde que possua medição de consumo energético e totalização de produção coletadas, dados que possuem características e estruturação similar aos utilizados neste trabalho.

Sugere-se também a possibilidade de explorar outras oportunidades não abordadas neste estudo, o que pode incluir:

- A incorporação de outros modelos nos *steps* 1 e 2 da arquitetura, ampliando as capacidades do da arquitetura para lidar com diferentes tipos de informações e contextos. Ao integrar modelos adicionais, é possível que surjam outros *insights*, contribuindo para uma tomada de decisão mais inteligente e eficaz no gerenciamento do consumo energético em ambientes industriais.
- A consideração de diferentes restrições no espaço de busca no *step* 3, alinhadas com as estratégias podem levar a soluções ainda mais otimizadas.
- A remodelagem da escala de tempo da base de dados, para outras possibilidades, além dos 15 minutos definidos para cada amostra, pode ser proporcionais novos resultados que permitam uma análise mais analítica dos padrões de consumo energético em diferentes escalas de tempo, como horas, dias ou semanas. Isso possibilita uma compreensão mais aprofundada das variações de consumo ao longo do tempo e facilita a identificação de padrões sazonais ou tendências relevantes para a gestão eficiente da energia.
- Cada *step* da arquitetura pode receber uma estrutura de aprendizado de máquina automatizado (do Inglês *Automated Machine Learning* - AutoML), possibilitando aos stakeholders com conhecimento limitado em ML realizar treinamentos e seleção dos

melhores modelos para compor a arquitetura que gera maior qualidade e eficiência nos processos para os quais ela for submetida.

Acredita-se que essas sugestões de novas explorações estão alinhadas com os desafios de inteligência artificial, ML, DL, eficiência energética e sustentabilidade, áreas promissoras para futuras investigações e desenvolvimentos na área de gestão energética industrial.

Referências

- ĐAKOVIĆ, D.; KLJAJIĆ, M.; MILIVOJEVIĆ, N.; DODER, Đ.; ANĐELKOVIĆ, A. S. Review of energy-related machine learning applications in drying processes. *Energies*, MDPI AG, v. 17, n. 1, p. 224, dez. 2023. Citado na página 17.
- APICELLA, A.; DONNARUMMA, F.; ISGRÒ, F.; PREVETE, R. A survey on modern trainable activation functions. *Neural Netw.*, Elsevier BV, v. 138, p. 14–32, jun. 2021. Citado na página 27.
- ASL, G. S.; SHAMSI, K.; THULASIRAM, R. K.; AKCOR, C.; LEUNG, C. Deep learning-based credit score prediction: Hybrid lstm-gru model. In: *2023 IEEE Symposium Series on Computational Intelligence (SSCI)*. [S.l.: s.n.], 2023. p. 395–400. Citado na página 34.
- BASHIR, T.; HAOYONG, C.; TAHIR, M. F.; LIQIANG, Z. Short term electricity load forecasting using hybrid prophet-LSTM model optimized by BPNN. *Energy Rep.*, Elsevier BV, v. 8, p. 1678–1686, nov. 2022. Citado na página 17.
- BHARATI, P.; PRAMANIK, A. Deep learning techniques—R-CNN to mask R-CNN: A survey. In: *Computational Intelligence in Pattern Recognition*. Singapore: Springer Singapore, 2020, (Advances in intelligent systems and computing). p. 657–668. Citado na página 26.
- BOUKTIF, S.; FIAZ, A.; OUNI, A.; SERHANI, M. A. Multi-sequence lstm-rnn deep learning and metaheuristics for electric load forecasting. *Energies*, MDPI AG, v. 13, n. 2, p. 391, jan. 2020. Citado na página 32.
- BREIMAN, L. *Mach. Learn.*, Springer Nature, v. 45, n. 1, p. 5–32, 2001. Citado na página 24.
- CAHUANTZI, R.; CHEN, X.; GÜTTEL, S. A comparison of LSTM and GRU networks for learning symbolic sequences. In: *Lecture Notes in Networks and Systems*. Cham: Springer Nature Switzerland, 2023, (Lecture notes in networks and systems). p. 771–785. Citado na página 34.
- CARVALHO, I. *Como funciona uma linha de montagem de automóveis?* 2017. Acesso em 20-03-2024. Disponível em: <<https://quatorrodas.abril.com.br/noticias/como-funciona-uma-linha-de-montagem-de-automoveis>>. Citado na página 16.
- CAVALCANTE, E. S.; VASCONCELOS, L. G. S.; NETO, G. W. de F.; RAMOS, W. B.; BRITO, R. P. Automotive painting process: Minimizing energy consumption by using adjusted convective heat transfer coefficients. *Prog. Org. Coat.*, Elsevier BV, v. 140, n. 105479, p. 105479, mar. 2020. Citado na página 17.
- CHEN, T.; GUESTRIN, C. XGBoost. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2016. Citado na página 24.

CHO, K.; MERRIENBOER, B. van; BAHDANAU, D.; BENGIO, Y. On the properties of neural machine translation: Encoder-decoder approaches. arXiv, 2014. Citado na página 33.

ELSAYED, N.; MAIDA, A.; BAYOUMI, M. Deep gated recurrent and convolutional network hybrid model for univariate time series classification. *Int. J. Adv. Comput. Sci. Appl.*, The Science and Information Organization, v. 10, n. 5, 2019. Citado na página 34.

FATAHI, R. Modeling of energy consumption factors for an industrial cement vertical roller mill by SHAP-XGBoost: a “conscious lab” approach. *approach. Scientific Reports*, v. 12, n. 1, 2022. Citado na página 17.

GIAMPIERI, A.; LING-CHIN, J.; MA, Z.; SMALLBONE, A.; ROSKILLY, A. P. A review of the current automotive manufacturing practice from an energy perspective. *Appl. Energy*, Elsevier BV, v. 261, n. 114074, p. 114074, mar. 2020. Citado na página 16.

HEWAMALAGE, H.; BERGMEIR, C.; BANDARA, K. Recurrent neural networks for time series forecasting: Current status and future directions. *Int. J. Forecast.*, Elsevier BV, v. 37, n. 1, p. 388–427, jan. 2021. Citado na página 26.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural Comput.*, MIT Press - Journals, v. 9, n. 8, p. 1735–1780, nov. 1997. Citado na página 31.

HOLLAND, J. H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis With Applications to Biology Control and Artificial Intelligence*. Ann Arbor, MI, USA: University of Michigan, 1975. Citado na página 35.

HONG, J.; LIANG, F.; YANG, H.; ZHANG, C.; ZHANG, X.; ZHANG, H.; WANG, W.; LI, K.; YANG, J. Multi- forward-step state of charge prediction for real-world electric vehicles battery systems using a novel LSTM-GRU hybrid neural network. *eTransportation*, Elsevier BV, v. 20, n. 100322, p. 100322, maio 2024. Citado na página 61.

HOVGARD, M.; LENNARTSON, B.; BENGTSSON, K. Energy-optimal timing of stochastic robot stations in automotive production lines. In: *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*. [S.l.]: IEEE, 2022. Citado na página 18.

HU, Y.; HUBER, A.; ANUMULA, J.; LIU, S.-C. Overcoming the vanishing gradient problem in plain recurrent networks. arXiv, 2018. Citado na página 27.

ILBEIGI, M.; GHOMEISHI, M.; DEHGHANBANADAKI, A. Prediction and optimization of energy consumption in an office building using artificial neural network and a genetic algorithm. *Sustain. Cities Soc.*, Elsevier BV, v. 61, n. 102325, p. 102325, out. 2020. Citado na página 18.

JIANG, P.; WANG, Z.; LI, X.; WANG, X. V.; YANG, B.; ZHENG, J. Energy consumption prediction and optimization of industrial robots based on LSTM. *J. Manuf. Syst.*, Elsevier BV, v. 70, p. 137–148, out. 2023. Citado 2 vezes nas páginas 17 e 23.

KATOCH, S.; CHAUHAN, S. S.; KUMAR, V. A review on genetic algorithm: past, present, and future. *Multimed. Tools Appl.*, Springer Science and Business Media LLC, v. 80, n. 5, p. 8091–8126, 2021. Citado na página 36.

KERVANCI, I. S.; AKAY, M. F.; ÖZCEYLAN, E. Bitcoin price prediction using LSTM, GRU and hybrid LSTM-GRU with bayesian optimization, random search, and grid search for the next days. *J. Ind. Manag. Optim.*, American Institute of Mathematical Sciences (AIMS), v. 20, n. 2, p. 570–588, 2024. Citado na página 63.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, Springer Science and Business Media LLC, v. 521, n. 7553, p. 436–444, maio 2015. Citado na página 26.

LU, Y.; SHENG, B.; FU, G.; LUO, R.; CHEN, G.; HUANG, Y. Prophet-EEMD-LSTM based method for predicting energy consumption in the paint workshop. *Appl. Soft Comput.*, Elsevier BV, v. 143, n. 110447, p. 110447, ago. 2023. Citado 2 vezes nas páginas 17 e 18.

LUKIN, E.; KRAJNOVIĆ, A.; BOSNA, J. Sustainability strategies and achieving SDGs: A comparative analysis of leading companies in the automotive industry. *Sustainability*, MDPI AG, v. 14, n. 7, p. 4000, mar. 2022. Citado na página 19.

MA, S.; JIANG, Z.; LIU, W. Modeling drying-energy consumption in automotive painting line based on ANN and MLR for real-time prediction. *Int. J. Precis. Eng. Manuf.-Green Technol.*, Springer Science and Business Media LLC, v. 6, n. 2, p. 241–254, abr. 2019. Citado na página 15.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, Springer Science and Business Media LLC, v. 5, n. 4, p. 115–133, dez. 1943. Citado na página 26.

NARCISO, D. A.; MARTINS, F. Application of machine learning tools for energy efficiency in industry: A review. *Energy Reports*, v. 6, p. 1181–1199, 2020. ISSN 2352-4847. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2352484719308686>>. Citado na página 18.

OKTAVIANDRI, M.; SAFIEE, A. S. B. Modelling electrical energy consumption in automotive paint shop. *IOP Conf. Ser. Mater. Sci. Eng.*, IOP Publishing, v. 319, p. 012060, mar. 2018. Citado na página 15.

OLIVEIRA, R. B. d.; SOUZA, P. H. Couto de L.; SILVA, T. C.; KIRSCHNER, V. E. S.; LORENZATO, F.; MACIEL, A. M. A. Análise de consumo de energia elétrica de equipamentos em oficina de pintura automotiva. *Rev. Eng. Pesqui. Apl.*, Revista de Engenharia e Pesquisa Aplicada, v. 9, n. 1, p. 69–78, dez. 2023. Citado 4 vezes nas páginas 42, 44, 49 e 61.

OPARA, K. R.; ARABAS, J. Differential evolution: A survey of theoretical analyses. *Swarm Evol. Comput.*, Elsevier BV, v. 44, p. 546–558, fev. 2019. Citado na página 38.

PASCANU, R.; GULCEHRE, C.; CHO, K.; BENGIO, Y. How to construct deep recurrent neural networks. arXiv, 2013. Citado na página 31.

PATRA, G. R.; MOHANTY, M. N. An LSTM-GRU based hybrid framework for secured stock price prediction. *J. Stat. Manag. Syst.*, Informa UK Limited, v. 25, n. 6, p. 1491–1499, ago. 2022. Citado na página 34.

RAGB, H.; MUNTASER, A.; JERA, E.; SAIDE, A.; ELWARFALLI, I. Hybrid GRU-LSTM recurrent neural network-based model for real estate price prediction. 2023. Citado na página 34.

- ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol. Rev.*, American Psychological Association (APA), v. 65, n. 6, p. 386–408, nov. 1958. Citado na página 26.
- RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. *Nature*, Springer Science and Business Media LLC, v. 323, n. 6088, p. 533–536, out. 1986. Citado na página 26.
- SZÁSZ, L.; CSÍKI, O.; RÁCZ, B.-G. Sustainability management in the global automotive industry: A theoretical model and survey study. *Int. J. Prod. Econ.*, Elsevier BV, v. 235, n. 108085, p. 108085, maio 2021. Citado na página 19.
- TELILI, A.; FEZZA, S. A.; HAMIDOUCHE, W.; MEFTAH, H. F. Z. B. 2BiVQA: Double Bi-LSTM based video quality assessment of UGC videos. *ACM Trans. Multimed. Comput. Commun. Appl.*, Association for Computing Machinery (ACM), nov. 2023. Citado na página 32.
- WANG, Z.; WANG, Y.; ZENG, R.; SRINIVASAN, R. S.; AHRENTZEN, S. Random forest based hourly building energy prediction. *Energy Build.*, Elsevier BV, v. 171, p. 11–25, jul. 2018. Citado 2 vezes nas páginas 17 e 38.
- YANG, J.; SUN, T.; HUANG, X.; PENG, K.; CHEN, Z.; QIAN, G.; QIAN, Z. Optimizing painting sequence scheduling based on adaptive Partheno-Genetic algorithm. *Processes (Basel)*, MDPI AG, v. 9, n. 10, p. 1714, set. 2021. Citado na página 15.
- YANG, S.; YU, X.; ZHOU, Y. LSTM and GRU neural network performance comparison study: Taking yelp review dataset as an example. In: *2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI)*. [S.l.]: IEEE, 2020. Citado na página 34.
- ZAFAR, N.; HAQ, I. U.; CHUGHTAI, J.-U.-R.; SHAFIQ, O. Applying hybrid lstm-gru model based on heterogeneous data sources for traffic speed prediction in urban areas. *Sensors (Basel)*, MDPI AG, v. 22, n. 9, p. 3348, abr. 2022. Citado 2 vezes nas páginas 17 e 63.
- ZHU, K.; YANG, X.; ZHANG, Y.; LIANG, M.; WU, J. A heterogeneity-aware car-following model: Based on the XGBoost method. *Algorithms*, MDPI AG, v. 17, n. 2, p. 68, fev. 2024. Citado na página 25.
- ZONG, M.; KRISHNAMACHARI, B. a survey on GPT-3. arXiv, 2022. Citado na página 26.