

INSTITUTO FEDERAL DO ESPÍRITO SANTO
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

FABRÍCIO RIBEIRO FERREIRA

**PREDIÇÃO DA TEMPERATURA DE DOMO EM UM REGENERADOR
UTILIZANDO ALGORITMOS DE APRENDIZADO SUPERVISIONADO**

Serra
2024

FABRÍCIO RIBEIRO FERREIRA

**PREDIÇÃO DA TEMPERATURA DE DOMO EM UM REGENERADOR
UTILIZANDO ALGORITMOS DE APRENDIZADO SUPERVISIONADO**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação Aplicada do Instituto Federal do Espírito Santo para obtenção do Título de Mestre em Computação Aplicada.

Orientador: Prof. Dr. Leandro Colombi Resendo

Serra
2024

Dados Internacionais de Catalogação na Publicação (CIP)

F383p
2024 Ferreira, Fabrício Ribeiro
Predição da temperatura de domo em um regenerador utilizando algoritmos de aprendizado supervisionado / Fabrício Ribeiro Ferreira. - 2024.
60 f.; il.; 30 cm

Orientador: Prof. Dr. Leandro Colombi Resendo.

Dissertação (mestrado) - Instituto Federal do Espírito Santo, Programa de Pós-graduação em Computação Aplicada, 2024.

1. Aprendizado de máquina. 2. Siderurgia. 3. K-Nearest neighbors (KNN). 4. Light gradient-boosting machine (LightGBM). 5. Hot blast stove I. Rosendo, Leandro Colombi. II. Instituto Federal do Espírito Santo. III. Título.

CDD 004

Bibliotecário: Valmir Oliveira de Aguiar - CRB6/ES 566

FABRÍCIO RIBEIRO FERREIRA

**Predição da Temperatura de Domo em um Regenerador utilizando algoritmos de
Aprendizado Supervisionado**

Dissertação apresentada ao Programa de Pós
Graduação em Computação Aplicada do
Instituto Federal do Espírito Santo, Campus
Serra, como requisito parcial para a obtenção
do título de Mestre em Computação Aplicada.

Aprovado em 20 de Dezembro de 2024

COMISSÃO EXAMINADORA

(Orientador Principal): Leandro Colombi Resendo
Instituto Federal do Espírito Santo

(Membro Interno): Thiago Meireles Paixão
Instituto Federal do Espírito Santo

(Membro Externo): Filipe Wall Mutz
Universidade Federal do Espírito Santo

Jade Barbosa Kill Lemos
(Membro Externo): Jade Barbosa Kill Lemos
ArcelorMittal

FOLHA DE ROSTO Nº 86/2024 - SER-CGEN (11.02.32.01.08.02)

(Nº do Protocolo: NÃO PROTOCOLADO)

(Assinado digitalmente em 23/12/2024 14:58)
LEANDRO COLOMBI RESENDO
PROFESSOR DO ENSINO BASICO TECNICO E TECNOLOGICO
SER-CGEN (11.02.32.01.08.02)
Matrícula: 1687072

(Assinado digitalmente em 24/12/2024 07:24)
THIAGO MEIRELES PAIXAO
PROFESSOR DO ENSINO BASICO TECNICO E TECNOLOGICO
SER-CGEN (11.02.32.01.08.02)
Matrícula: 1904995

Visualize o documento original em <https://sipac.ifes.edu.br/documentos/> informando seu número: **86**, ano: **2024**,
tipo: **FOLHA DE ROSTO**, data de emissão: **23/12/2024** e o código de verificação: **16d3e6aa28**

AGRADECIMENTOS

À minha esposa Tânia, minha grande incentivadora, que caminhou ao meu lado com incansável apoio e motivação, e à minha filha Ana Carolina, pela compreensão e paciência diante das ausências necessárias durante este desafiador período.

Aos meus pais, Núbia e Dorval, por sua educação e constante incentivo aos estudos, muitas vezes abrindo mão de seus próprios desejos em prol da nossa formação. Aos meus irmãos Júnior e Flávia, meus cunhados e sobrinhos, que me acompanharam nesta jornada, e aos meus avós Creuza e Ilis (in memoriam), por seu legado de amor e sabedoria.

A Deus, pela saúde, auxílio e direção, e à minha igreja, pelas orações e apoio nos momentos mais difíceis e a compreensão pelas ausências.

Ao meu orientador, Professor Doutor Leandro Colombi Resendo, pela dedicação, apoio inabalável, confiança e competência na orientação deste trabalho, que foram fundamentais para a superação dos desafios enfrentados.

Aos membros da banca examinadora, por gentilmente aceitarem o convite e dedicarem seu tempo e conhecimento à análise deste trabalho.

Aos colegas do Mestrado Profissional em Computação Aplicada, pelo companheirismo, pelas conversas enriquecedoras, pelas ajudas mútuas e pela troca de experiências, que criaram um ambiente saudável de convivência e aprendizado.

Aos demais professores e funcionários do Instituto Federal do Espírito Santo - Campus Serra, a FAPES e CAPES (processo 2021-2S6CD, nº FAPES 132/2021) por meio do PDPG (Programa de Desenvolvimento da Pós-Graduação, Parcerias Estratégicas nos Estados) por proporcionarem uma estrutura acolhedora e eficiente para o desenvolvimento deste trabalho.

Resumo

Este trabalho realiza estudo comparativo de técnicas de aprendizado de máquina e analisa a aplicação de diversos modelos de aprendizado de máquina na predição da temperatura do domo de regeneradores durante os ciclos de *on-gas* e *on-blast* em uma usina siderúrgica integrada, visto que o consumo de gases durante o período de queima (*on-blast*) e aquecimento dos tijolos refratários (*on-gas*), não apenas impacta diretamente nos custos operacionais, mas também acarreta significativas implicações ambientais, e a predição de temperatura do domo de um regenerador permitirá atualizar adequadamente o plano de ação, reajustando as variáveis para atingir a temperatura de domo alvo com maior eficiência energética. O modelo utiliza os valores das variáveis de entrada, definidos pelos operadores e coletadas pelos sensores, para prever a temperatura do domo após 36 minutos, dado o *setup* atual. Diferentemente de todos os trabalhos encontrados na literatura, as temperatura de domo do passado registrada na base de dados nesta investigação não permitem prever a temperatura do domo no futuro, o que impossibilita o uso de algoritmos especializados que consideram essa característica. Utilizando uma base de dados fornecida por uma siderúrgica no Brasil, inicialmente foi realizada uma pré-seleção de atributos por meio de entrevistas com especialistas, seguida do uso do método de Análise de Componentes Principais (PCA). O pré-processamento incluiu a remoção de registros duplicados e de *outliers* usando o Intervalo Interquartil (IRQ) e a imputação de valores faltantes com a média móvel de uma janela fixa que considerou as 24 amostras anteriores, a técnica *Min-Max Scaling* foi aplicada para normalizar os valores, transformando as variáveis do conjunto para a escala. Para o treinamento dos modelos, foi utilizado o método de validação cruzada, dividindo a base em 80% dos dados para treino e 20% para teste. Os modelos foram avaliados com métricas como R^2 e RMSE, após ajustes de hiperparâmetros via *Grid Search CV* e *k-fold* de 5. O LightGBM e o XGBoost apresentaram, respectivamente, os melhores resultados para R^2 de 0,87040695 e 0,83274270 e RMSE de 3,0538916 e 3,2550260. Os resultados indicam uma eficácia significativa dos modelos, com uma ligeira vantagem do LightGBM.

Palavras-chave: Hot Blast Stove. HBS. Random Forests. Gradient Boosted. LightGBM. XGBoost. KNN.

ABSTRACT

This work conducts a comparative study of machine learning techniques and analyzes the application of various machine learning models in predicting the dome temperature of regenerators during the on-gas and on-blast cycles in an integrated steel plant. Unlike studies found in the literature, the data used in this investigation do not exhibit time series behavior, which precludes the use of specialized algorithms that consider this characteristic. Using a dataset provided by a steel plant in Brazil, an initial attribute pre-selection was conducted through interviews with experts, followed by the use of the Principal Component Analysis (PCA) method. The preprocessing included the removal of duplicate records and outliers using the Interquartile Range (IQR) and the imputation of missing values with a moving average of a fixed window that considered the previous 24 samples. The Min-Max Scaling technique was applied to normalize the values, transforming the variables to the same scale. For model training, cross-validation was used, dividing the dataset into 80% for training and 20% for testing. The models were evaluated using metrics such as R^2 and RMSE, after hyperparameter tuning via GridSearchCV and 5-fold cross-validation. The LightGBM and XGBoost models showed the best results, with R^2 values of 0.87040695 and 0.83274270, and RMSE values of 3.0538916 and 3.2550260, respectively. The results indicate significant model efficacy, with a slight advantage for LightGBM.

Keywords: Hot Blast Stove. HBS. Random Forests. Gradient Boosted. LightGBM. XGBoost. KNN.

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1 – Período de <i>on-gas</i> e <i>on-blast</i> . Fonte: | 12 |
| Figura 2 – Funcionamento do HBS nas fases <i>on-gas</i> e <i>on-blast</i> | 13 |
| Figura 3 – <i>ViolinPlot</i> para comparar os dados antes e depois do pré-processamento. | 29 |
| Figura 4 – KNN classificação. | 32 |
| Figura 5 – Exemplo de Matriz de Confusão. | 34 |
| Figura 6 – Limite de decisão na separação do espaço e classe. | 37 |
| Figura 7 – SVM- Classificação Não Linear. | 37 |
| Figura 8 – Princípio básico do <i>bagging</i> | 40 |
| Figura 9 – Diagrama esquemático do algoritmo do histograma. | 43 |
| Figura 10 – LightGBM pesquisa com crescimento da árvore por folha. | 44 |
| Figura 11 – Diagrama esquemático do XGBoost. | 49 |
| Figura 12 – Comparação R^2 e RMSE dos testes antes e depois dos hiperparâmetros. | 53 |
| Figura 13 – Gráficos de dispersão comparando a predição dos métodos XGBoost e LightGBM. | 54 |
| Figura 14 – Comparação da predição dos métodos XGBoost e LightGBM | 55 |

LISTA DE SIGLAS

| | |
|----------------|--|
| AF | - Alto-forno |
| BFG | - <i>Blast Furnace Gas</i> |
| CNN | - <i>Convolutional Neural Network</i> |
| COG | - <i>Coke Gas</i> |
| COP21 | - Conferência das Partes |
| DT | - <i>Decision Tree</i> |
| DMESN | - <i>Deep Memory Echo State Network</i> |
| ESN | - <i>Echo State Network</i> |
| GBDT-MM | - <i>Gradient Boosting Decision Tree</i> |
| GBM | - <i>Gradient Boosted Machines</i> |
| HBS | - <i>Hot Blast Stoves</i> |
| IA | - Inteligência Artificial |
| IQR | - <i>Interquartile Range</i> |
| KNN | - <i>K-Nearest Neighbors</i> |
| LightGBM | - <i>Light Gradient Boosting Machine</i> |
| LSTM | - <i>Long Short-Term Memory Neural Network</i> |
| MAPE | - <i>Mean Absolute Percentual Error</i> |
| MdRAE | - <i>Median Relative Absolute Error</i> |
| ML | - <i>Machine Learning</i> |
| MLP | - <i>MultiLayer Perceptron</i> |
| MSE | - <i>Mean Squared Error</i> |
| NN | - <i>Neural Networks</i> |
| PCA | - <i>Principal Component Analysis</i> |
| R ² | - <i>R Squared</i> |
| RF | - <i>Random Forest</i> |

| | |
|---------|--------------------------------------|
| RMSE | - <i>Root Mean Squared Error</i> |
| RNN | - <i>Recurrent Neural Network</i> |
| SGD | - <i>Stochastic Gradient Descent</i> |
| SVM | - <i>Support Vector Machine</i> |
| SVR | - <i>Support Vector Regression</i> |
| VAR | - <i>Vector Autoregression</i> |
| XGBoost | - <i>Extreme Gradient Boosting</i> |

SUMÁRIO

| | | |
|--------------|---|-----------|
| 1 | INTRODUÇÃO | 11 |
| 1.1 | PROPOSTA DO TRABALHO | 15 |
| 1.2 | OBJETIVO GERAL | 16 |
| 1.3 | OBJETIVOS ESPECÍFICOS | 16 |
| 1.4 | ORGANIZAÇÃO DO TRABALHO | 16 |
| 2 | TRABALHOS CORRELATOS | 17 |
| 2.1 | PREDIÇÃO DE TEMPERATURA | 17 |
| 2.2 | OPERAÇÃO DE REGENERADORES (HBS) | 19 |
| 2.3 | COMPARAÇÃO DE MODELOS | 21 |
| 3 | MATERIAIS E MÉTODOS | 26 |
| 3.1 | COLETA DE DADOS | 26 |
| 3.2 | SELEÇÃO DE ATRIBUTOS | 27 |
| 3.3 | PRÉ-PROCESSAMENTO DOS DADOS | 27 |
| 3.4 | MODELOS INVESTIGADOS | 30 |
| 3.4.1 | K-Nearest Neighbors (KNN) | 30 |
| 3.4.1.1 | Armazenamento dos Dados | 31 |
| 3.4.1.2 | Escolha do Parâmetro k e Identificação dos k Vizinhos Mais Próximos | 31 |
| 3.4.1.3 | Classificação e Regressão | 32 |
| 3.4.1.4 | Métricas de desempenho | 34 |
| 3.4.2 | Support Vector Machines (SVM) | 36 |
| 3.4.3 | Random Forests (RF) | 38 |
| 3.4.3.1 | Árvores de Decisão | 38 |
| 3.4.3.2 | Bagging ou Bootstrap aggregation | 39 |
| 3.4.4 | Gradient Boosting Machines | 40 |
| 3.4.4.1 | Light Gradient Boosting Machine (LightGBM) | 42 |
| 3.4.4.2 | Extreme Gradient Boosting (XGBoost) | 46 |
| 3.5 | MÉTRICAS | 48 |
| 4 | RESULTADOS E DISCUSSÕES | 52 |
| 5 | CONCLUSÕES E TRABALHOS FUTUROS | 56 |
| | REFERÊNCIAS | 57 |

1 INTRODUÇÃO

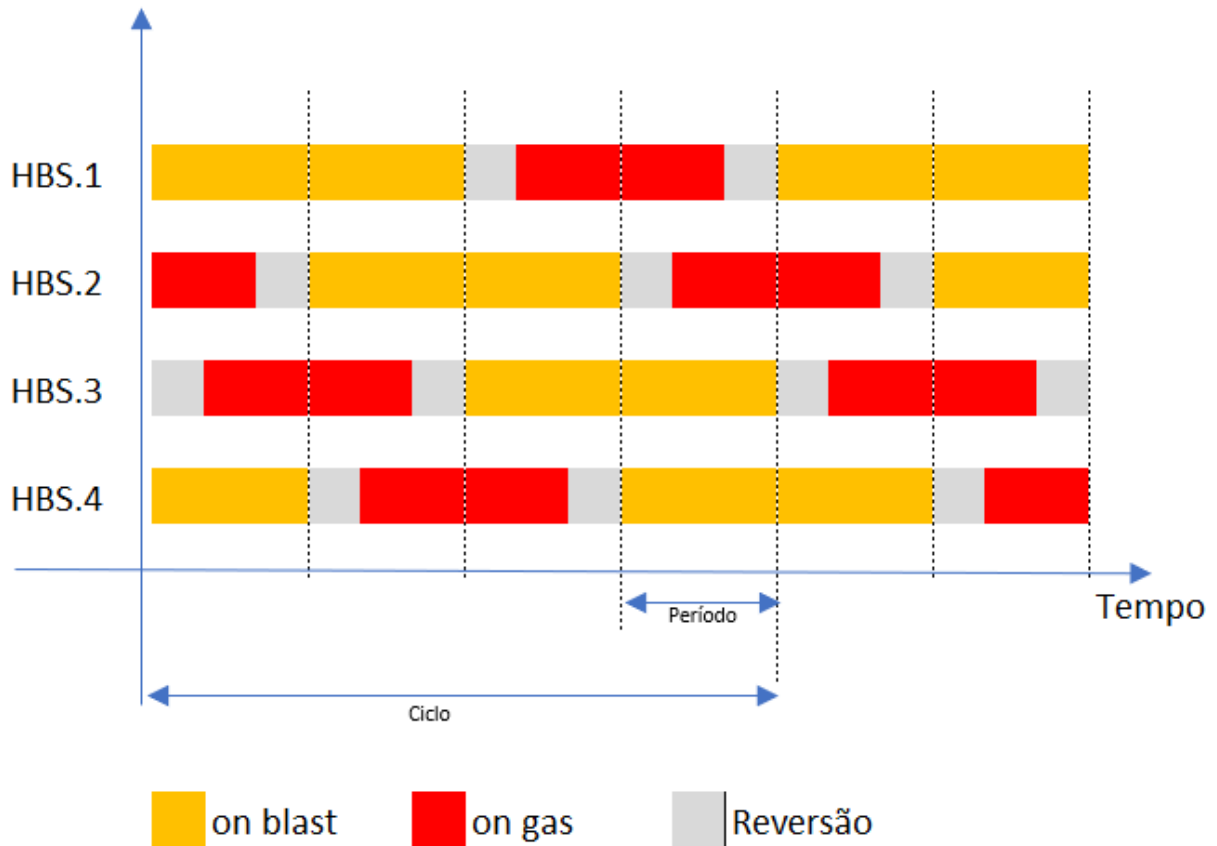
A Conferência das Partes da Convenção-Quadro das Nações Unidas sobre Mudança do Clima (COP21), realizada em Paris em 2015, estabeleceu um acordo histórico entre os governos para enfrentar o desafio das mudanças climáticas e mitigar o impacto do aumento da temperatura global (OBERGASSEL et al., 2015). Como parte desse acordo, a União Europeia assumiu o compromisso de reduzir em pelo menos 40% as emissões de gases de efeito estufa até 2030. Essa meta é particularmente crucial para a indústria siderúrgica, que desempenha um papel proeminente nas emissões globais de gases chegando a representar até 6,7% do total de emissões e 31% das emissões industriais de dióxido de carbono CO₂ (SHATOKHA, 2016).

A siderurgia tem grande representatividade na economia do Brasil. De acordo com o Instituto Aço Brasil (BRASIL, 2021), a indústria brasileira do aço foi responsável por 1,7% da produção mundial em 2020, posicionando o país na 9ª colocação no ranking mundial de produtores do setor. Em 2023, a contribuição do país foi de 1,8% da produção mundial, mantendo-se na 9ª posição (BRASIL, 2023). Na América Latina, o Brasil manteve-se na 1ª posição, com 56,1% e 54,9% da produção da região respectivamente em 2020 e 2023. No ano de 2020, a produção de ferro-gusa foi de aproximadamente 24.517.000 toneladas enquanto a produção de aço bruto e de laminados foram de aproximadamente 30.971.000 e 21.664.000 toneladas, respectivamente (BRASIL, 2020). Em 2023, a produção de aço bruto foi de aproximadamente 34.1 milhões de toneladas (BRASIL, 2023).

O processo de fabricação de ferro em um Alto-Forno (AF) é o cerne do processo tradicional de fabricação de aço, representando cerca de 65% do consumo total de energia na indústria siderúrgica (ZHOU; ZHANG; YANG, 2019). De acordo com Sun, Wang e Wang (2020), a produção de aço é responsável por 5% do consumo global de energia e por uma grande parte das emissões globais de carbono. Visando reduzir os custos ligados ao consumo de energia e diminuir as emissões de CO₂, a indústria siderúrgica tem se empenhado em encontrar soluções inovadoras, novas fontes de energia e otimizações nos processos.

Dentre os processos de otimização na produção de aço, destaca-se a reutilização de subprodutos do processo siderúrgico, em especial o gás de alto-forno (BFG, do inglês *Blast Furnace Gas*). Este subproduto é valorizado pelo seu volume de produção, e sua capacidade calorífica, aproximadamente 17.580 ~ 18.420 MJ/m³ (SUI et al., 2023), juntamente com o gás de coqueria (COG), que possui valor calorífico de 16,23 MJ/m³ (ORTIZ-IMEDIO; ORTIZ; ORTIZ, 2022). A reutilização desses gases residuais, especialmente o BFG, é essencial para aumentar a eficiência energética do processo siderúrgico e reduzir a emissão de CO₂. O BFG, que contém cerca de 9 ~ 12% em peso de CO₂, possui uma intensidade de carbono maior comparada a outros gases industriais e de usinas de energia (WU et al., 2021). O BFG é gerado durante a produção do aço. O minério de ferro reage com

Figura 1 – Período de *on-gas* e *on-blast*. Fonte:

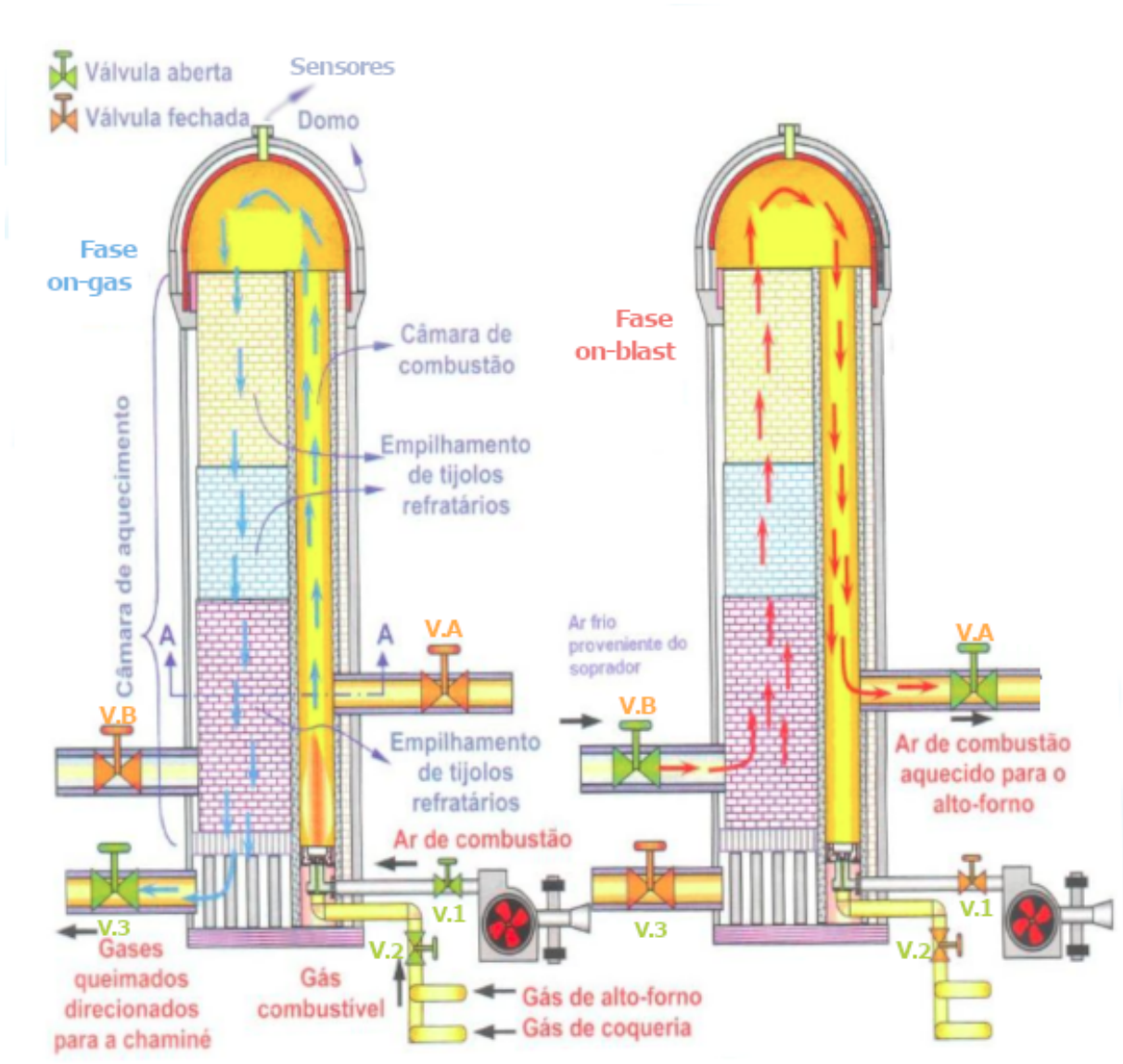


Fonte: (ŞAHIN; MORARI, 2008) e adaptado pelo Autor

carbono e oxigênio em alta temperatura dentro do alto-forno (AF) e reaproveitado em outros equipamentos da planta, como por exemplo nos regeneradores (HBS, do inglês *Hot Blast Stoves*) para reaquecer o ar que é uma prática fundamental para otimizar o uso de recursos energéticos, diminuir os custos operacionais e reduzir significativamente as emissões de gases de efeito estufa. Portanto, a adequada utilização de gases como o BFG e o COG não apenas melhora a eficiência do processo, mas também contribui de maneira importante para a redução das emissões unitárias de CO₂ na produção de aço.

Os HBS ou regeneradores, são fundamentais na siderurgia. Tais equipamentos são responsáveis por fornecer ar em altas temperaturas, aproximadamente 1.200°C, necessário para o processo de produção do aço nos AF (KOYFMAN et al., 2021). Os HBS trabalham em paralelo e em ciclos, ou seja, enquanto um regenerador sopra o ar quente para o AF, perdendo calor ao longo da fase de *on-blast*, o(s) outro(s) são aquecidos. Em geral, são necessários entre 2 a 4, regeneradores trabalhando juntos, para manter uma produção constante de aço. A Figura 1, demonstra graficamente os ciclos de transição entre as fases de *on-gas* e *on-blast* ao longo do tempo, representando a troca das fases em uma planta com 4 HBS. Podemos observar que as operações são constantes e periodicamente alteradas

Figura 2 – Funcionamento do HBS nas fases *on-gas* e *on-blast*.



Fonte: (RIZZO, 2009) e adaptado pelo autor

entre as fases de *on-gas* representadas pelos retângulos amarelos e a fase de *on-blast*, representada pelos retângulos vermelhos, que por sua vez também têm duração constante. Observamos também que as operações de cada HBS são deslocadas no tempo em relação aos outros HBS. Esse deslocamento objetiva manter a constância de sopro de ar quente ao AF. Um ciclo de operação é a duração entre o início da fase de *on-gas* até o fim da fase de *on-blast* de um HBS, iniciando assim um novo ciclo para aquele HBS. Entre as fases de *on-gas* e *on-blast* há um período determinado de reversão, no qual ocorre a transição entre as fases e a alternância do fluxo de ar e das variáveis do processo representado pelo retângulo cinza. Na Figura 1, identifica-se um tempo denominado de período, que é a faixa de tempo em que 2 HBS's fornecem ar quente ao AF, enquanto 2 HBS's mantêm a frequência de ar frio no valor mínimo (ŞAHIN; MORARI, 2008).

Na Figura 2, é representado o processo de operação de dois regeneradores funcionando em paralelo, operando em fases diferentes, o primeiro em azul representa a fase de *on-gas* e o segundo em vermelho a fase de *on-blast*. O HBS da Figura 2 o regenerado que representa a fase *on-gas*, demonstra o processo de aquecimento, com a chama acesa em sua base. As válvulas controlam o fluxo de ar de combustão (NG) através da válvula V.1, assim como o gás de coqueria (COG) e o gás de alto-forno (BFG) são controlados através da válvula V.2, formando o gás de mistura que alimenta a chama. Nesta fase, a válvula V.A em direção ao alto-forno, e a válvula V.B de entrada do ar frio proveniente do soprador permanecem fechadas. O gás de mistura queimado se torna um fluxo de gás em alta temperatura, representado pela setas azuis, que percorre a câmara de combustão do HBS subindo até o topo cilíndrico, denominado domo.

No domo os sensores fazem aferições, dentre elas a temperatura do gás queimado. Seu formato cilíndrico direciona o fluxo de gás em alta temperatura para a câmara de aquecimento, onde estão os tijolos refratários empilhados de maneira que o fluxo do gás queimado passe por eles, realizando uma troca térmica e aquecendo os refratários. Na base da câmara de aquecimento, há a válvula V.3 que direciona o fluxo do gás queimado, após a passagem pelos tijolos refratários e a realização da troca térmica, para a chaminé. Durante a fase de *on-gas* a válvula V.3 permanece aberta.

Já a fase a Figura 2 fase *on-blast*, as válvulas V.1, V.2 e V.3, que controlam o fluxo de NG, BFG, COG e a saída de gases para chaminé estão fechadas. Enquanto as válvulas V.A e V.B, que controlam respectivamente o ar frio proveniente do soprador e a saída de ar de combustão para o alto-forno, estão abertas. O fluxo de ar frio proveniente do soprador é injetado no HBS através da válvula V.B na base da câmara de combustão, representado pelas setas vermelhas, passando pelos tijolos refratários que foram aquecidos durante a fase de *on-gas*. A troca térmica ocorre transformando o ar frio em ar de alta-temperatura, que segue em direção ao domo, onde a temperatura do ar e outras variáveis são aferidas pelos sensores. No domo o fluxo de ar em alta temperatura é direcionado para câmara de combustão e, conseqüentemente para a saída através da válvula de ar de combustão que direciona todo o fluxo para o alto-forno (ŞAHIN; MORARI, 2008).

O consumo de gases durante o período de queima e aquecimento dos tijolos refratários, ou seja, no período de *on-gas*, não apenas impacta diretamente nos custos operacionais, mas também acarreta significativas implicações ambientais. Portanto, a capacidade de prever temperatura do domo do regenerador permitirá ao operador atualizar adequadamente o plano de ação, reajustando as variáveis de entrada para atingir a temperatura de domo alvo com maior eficiência energética. Além disso, esse controle aprimorado do processo, proporcionado pela predição da temperatura de domo, também contribui para a prolongar a vida útil dos regeneradores, reduzir as paradas para manutenção e otimizar a utilização

dos recursos (ŞAHIN; MORARI, 2008; KWAKERNAAK; TIJSSEN; STRIJBOS, 1970).

1.1 PROPOSTA DO TRABALHO

Este trabalho propõe a investigação de métodos de predição da temperatura do domo do sistema regenerador HBS, utilizando técnicas de aprendizado de máquina, como um primeiro passo em direção à otimização do consumo de gases. Diversos modelos de aprendizado de máquina serão analisados quanto ao seu desempenho na previsão da temperatura do regenerador. Entre os algoritmos a serem comparados, estão: *Support Vector Machine* (SVM), um algoritmo de regressão supervisionada baseado em otimização convexa; *Random Forest* (RF); dois algoritmos de *Gradient Boosting*, sendo eles *Extreme Gradient Boosting* (XGBoost) e *Light Gradient Boosting Machine* (LightGBM); e *K-Nearest Neighbors* (KNN), um algoritmo baseado em instâncias que classifica novos pontos de dados com base na proximidade com os pontos de treinamento.

Embora diversos artigos sobre predição em ambientes siderúrgicos utilizem métodos específicos de *Machine Learning* e séries temporais, como mencionado em Navarro et al. (2022), e o estudo de Ding et al. (2018) que investigavam a aplicação de modelo para série temporais para modelagem e previsão do conteúdo de silício metálico quente. Testes preliminares indicaram que algoritmos tradicionais para predição em séries temporais regulares como aqueles baseados em ARIMA não demonstraram uma boa performance com a base de dados apresentada. Portanto, optamos por utilizar algoritmos de aprendizado de máquina. Os métodos investigados foram avaliados usando dados reais de uma empresa siderúrgica, e os resultados mostraram que o sistema é capaz de prever a temperatura do domo do regenerador com aproximadamente 30 minutos de antecedência. As métricas de comparação dos modelos são: a raiz do erro médio quadrático (*Root Mean Square Error* - RMSE), o coeficiente de determinação R^2 (*R squared*), erro absoluto médio (*Mean Absolute Error* - MAE) e erro percentual absoluto médio (*Mean Absolute Percentage Error* - MAPE) sendo as duas últimas, métricas utilizadas para avaliar o modelo selecionado. Um bom sistema de previsão de temperatura do domo permitirá planejar as ações de controle com antecedência, a fim de reduzir o consumo de gases utilizado na queima para aquecimento do HBS e aumentar a segurança da planta com a prevenção de acidentes. Além disso, os resultados de comparação dos modelos SMV, RF, XGBoost, LightGBM e KNN apresentaram erro médio de 18,84°C , 7,33°C, 4,91°C, 4,36°C, e 5,66°C respectivamente.

Este trabalho apresenta duas contribuições principais: (i) apresenta uma abordagem distinta da comumente encontrada na literatura, ao não considerar séries temporais para predição, conforme será explicado e justificado nas análises dos dados; e (ii) oferece uma comparação abrangente entre vários métodos de aprendizado de máquina, fornecendo *insights* sobre as diferentes abordagens para a previsão da temperatura do domo em sistemas regeneradores na indústria siderúrgica.

1.2 OBJETIVO GERAL

O objetivo principal deste trabalho é investigar os modelos para identificar o melhor algoritmo de aprendizado de máquina para realizar a predição da temperatura do domo de um regenerador (HBS), utilizando dados de sensores e das ações dos operadores de regeneradores de uma siderurgia em operação.

1.3 OBJETIVOS ESPECÍFICOS

Como objetivos específicos pretende-se realizar:

- Levantamento e análise dos dados disponíveis.
- Estudo bibliográfico para identificar os métodos de predição e as métricas de análise de algoritmos mais utilizados na literatura.
- Identificação e pré-seleção de atributos na base de dados por meio de entrevistas com especialistas.
- Aplicação da Análise de Componentes Principais (PCA) para definir os atributos mais relevantes da base de dados.
- Pré-processamento da base de dados realizando a remoção de registros duplicados e outlier, normalização e tratamento registros ausentes.
- Realização de experimentos para avaliar os algoritmos de predição.
- Comparação qualitativa e quantitativa dos resultados dos algoritmos.

1.4 ORGANIZAÇÃO DO TRABALHO

O restante do trabalho está organizado da seguinte forma: no Capítulo 2, é apresentada uma revisão de trabalhos correlatos recentes. O Capítulo 3, descreve o problema, o conjunto de dados e os tratamentos realizados nos dados. No Capítulo 4, são apresentados os modelos e, os experimentos, bem como os resultados numéricos das análises de predição. No Capítulo 5, são discutidas as considerações finais do trabalho realizado até o momento e os trabalhos futuros.

2 TRABALHOS CORRELATOS

2.1 PREDIÇÃO DE TEMPERATURA

A otimização de recursos é, provavelmente, um dos temas mais investigados nas indústrias siderúrgicas. Na busca por melhorias nos processos, uma das principais estratégias adotadas é a predição de estados em um sistema. Nesse sentido, a predição de temperatura nos diversos estágios da fabricação do aço é um dos temas mais relevantes. Devido aos custos envolvidos, aos riscos no processo e ao impacto na qualidade do produto principal, que é o ferro gusa, o cenário mais amplamente investigado costuma ser o do AF.

Bing-yao, Hui e You-jun (2011) propõem um modelo de controle que melhora a precisão da predição da temperatura final de siderúrgicas de pequeno e médio porte na China, reduzindo a necessidade de ajustes manuais e reoperações, que impactam negativamente na qualidade do aço e nos custos de produção. O modelo baseado em redes neurais artificiais, especificamente a Rede Neural de Função de Base Radial (RBF), conhecida por sua capacidade de aproximação não linear e rápida convergência para lidar com a alta presença de ruídos nos dados coletados do ambiente metalúrgico. O trabalho propõe a combinação do algoritmo de *clustering* DBSCAN com a RBF. O DBSCAN é utilizado para identificar e tratar objetos ruidosos, removendo outliers evidentes e reduzindo o impacto de valores marginais no treinamento da rede neural. Foram coletados 500 conjuntos de dados de uma siderúrgica, dos quais 330 foram usados para treinamento e 194 para teste. O modelo utilizou oito variáveis principais como entrada, incluindo quantidade de ferro-gusa, sucata, oxigênio, cal, dolomita, minério de ferro, bolas de escória e tempo de sopro de oxigênio. Após a normalização dos dados, o DBSCAN foi aplicado em duas etapas: a primeira para remover ruídos evidentes e a segunda para ajustar pesos de objetos marginais. A rede RBF foi então treinada com os dados processados, utilizando 40 neurônios na camada intermediária e o algoritmo de gradiente descendente para ajuste de pesos. Os experimentos compararam o desempenho da RBF básica com a RBF aprimorada. Os resultados mostraram que a RBF aprimorada teve melhor desempenho, com 135 acertos dentro de 15°C (contra 131 da RBF básica) e 101 acertos dentro de 10°C (contra 92 da RBF básica). O MSE (do inglês *Mean Squared Error*) foi reduzido de 15,65 para 14,95, e a soma MSE caiu de 47.523 para 43.337. Esses resultados demonstram que a abordagem proposta melhora a precisão da predição da temperatura final no BOF, tornando o modelo mais robusto e aplicável em cenários industriais com dados ruidosos.

O trabalho de Tang e Liu (2015) propõem um sistema de controle de temperatura para HBS a carvão, utilizando controle preditivo baseado em redes neurais. O objetivo da pesquisa é melhorar a precisão estática e as características dinâmicas do controle de temperatura em sistemas caracterizado por grande atraso e inércia, superando as limitações dos métodos clássicos de modelagem matemática. A metodologia empregada combina a teoria de

controle preditivo com um modelo de rede neural com três camadas (entrada, oculta e saída). A camada oculta utiliza a função *Sigmoid* e a camada de saída utiliza a função logarítmica S, a rede é treinada com amostras de temperatura e fornecimento de ar, normalizadas no intervalo $[0, 1]$, ou seja, trata-se de uma rede neural de retropropagação (BP) utilizada para prever a temperatura do HBS com base em dados de entrada e saída anteriores. O sistema é composto pelo objeto controlado, no caso o HBS, um modelo preditivo da rede neural, otimização contínua e correção de *feedback*. Foram realizadas simulações para testar a viabilidade do esquema de controle, com um valor de temperatura alvo de 700°C . A precisão da predição foi avaliada utilizando o erro médio quadrático e a acurácia, determinando o tamanho das camadas de 5-9-1 na estrutura de rede é a mais adequada, pois apresentou o menor erro de previsão. Os resultados mostraram que o sistema controlado pode seguir a trajetória de referência com pouca oscilação e sem ultrapassagem significativa, estabilizando rapidamente. Em aplicações práticas, os dados medidos confirmaram que os resultados previstos eram consistentes com as mudanças reais de temperatura, concluiu-se que o esquema de controle preditivo baseado em redes neurais é robusto, adaptável e superior aos métodos convencionais de controle PID.

O estudo de Kacur et al. (2022) apresenta um comparativo sobre a aplicação de métodos de aprendizado de máquina para a predição da temperatura e da concentração de carbono no processo de fabricação de aço em BOF. A ausência de medições contínuas da temperatura e da concentração de carbono durante a fusão representa um desafio significativo para o controle do processo. O trabalho utiliza modelos matemáticos e de aprendizado de máquina para lidar com as não linearidades das variáveis do processo, destacando a relevância de sensores virtuais para estimativas em tempo real. O trabalho envolveu a comparação de cinco técnicas de aprendizado de máquina: Multivariate Adaptive Regression Splines (MARS), Support Vector Regression (SVR), Redes Neurais (NN), k-Nearest Neighbors (k-NN) e Random Forest (RF). Os experimentos foram realizados com dados estáticos e dinâmicos de múltiplas fusões, sendo proposto um método de pareamento entre esses dois tipos de dados para a criação de conjuntos de treinamento. As métricas utilizadas para avaliação incluíram indicadores estatísticos, como erro absoluto médio (MAE) e raiz do erro quadrático médio (RMSE), além do índice de desempenho (PI) e do tempo de treinamento. Os resultados indicaram que o modelo MARS, com abordagem cúbica por partes, apresentou o melhor desempenho na predição da temperatura no teste com dados estáticos e dinâmicos, alcançando um PI de 0,50. Para a predição da concentração de carbono, o modelo k-NN foi o mais preciso, enquanto as Redes Neurais demonstraram o pior desempenho em ambos os casos, com PI de 0,81. Em termos de tempo de treinamento, o k-NN foi o mais rápido (0,001 s), enquanto o RF foi o mais lento (64,58 s). A análise também revelou que a predição da temperatura foi mais eficiente com dados estáticos, enquanto a predição de carbono foi mais precisa com dados combinados.

2.2 OPERAÇÃO DE REGENERADORES (HBS)

Na década de 1970, estudos iniciais sobre a operação ideal de regeneradores (HBS) em operação paralela escalonada foram realizados por Kwakernaak, Tijssen e Strijbos (1970) com foco na eficiência térmica, dado que o consumo de combustível representa cerca de metade dos custos operacionais dos regeneradores. O trabalho comparou a eficiência do sistema paralelo escalonado com a operação em série de quatro regeneradores, desenvolvendo uma abordagem numérica que envolveu a manipulação de variáveis como a taxa de fluxo de gás quente e a temperatura de entrada do gás e a duração dos períodos de aquecimento e resfriamento. Os experimentos utilizaram um algoritmo do tipo *quasi-Newton* implementado em Algol para mostrar que a operação paralela escalonada é mais eficiente termicamente e menos sensível a variações na duração do período de resfriamento em comparação com a operação em série. Nesse experimento, foram medidas a temperatura final do ar quente, a temperatura dos tijolos refratários e o volume de mistura de ar frio, com o objetivo de minimizar o consumo de energia e maximizar a eficiência térmica. A pesquisa concluiu que a operação paralela escalonada é consideravelmente mais eficiente e robusta, economizando custos operacionais e reduzindo a necessidade de ajustes frequentes.

No estudo realizado por Şahin e Morari (2008), é proposto um esquema de controle otimizado para minimizar o consumo de energia nos HBS em operação paralela escalonada. A abordagem busca reduzir o fornecimento de calor aos fornos enquanto mantém a temperatura e a taxa de fluxo do ar quente necessária para o AF. O problema de minimização de energia foi definido como um problema de controle ótimo com restrições. Para reduzir o consumo de energia, foi utilizado um esquema de controle preditivo de modelos (MPC), que permite trabalhar com sistema com multivariáveis e incorporar restrições operacionais. O modelo de controle foi desenvolvido a partir de experimentos de resposta ao degrau em um modelo dinâmico detalhado, traduzindo a operação periódica e variável no tempo dos fornos em uma dinâmica invariante no tempo, simplificando o problema. Este modelo descreve a transferência de calor dentro dos fornos, e considera suposições comuns como a negligência das perdas de calor para o ambiente e a capacidade de calor do gás ser insignificante em comparação à do material refratário. A simulação do esquema MPC foi realizada em um modelo dinâmico detalhado, com a medição de variáveis como a temperatura final do ar quente e a temperatura dos tijolos de sílica. A resposta do sistema ao controle de fluxo de gás de exaustão e a manutenção das temperaturas dentro dos limites estabelecidos também foram avaliadas. Os resultados indicaram uma redução no tempo para atingir o ponto de operação ótimo em aproximadamente 200 períodos (50 ciclos), o que é quatro vezes mais rápido que a resposta em malha aberta. O MPC alcançou a temperatura final do jato em 150 períodos com um desvio de $0,4^{\circ}\text{C}$, sem violar a restrição de temperatura dos tijolos, sendo mantida dentro dos limites com uma violação

máxima de 2°C. O controle preditivo também conseguiu lidar com mudanças nas restrições ativas durante a operação, demonstrando a eficácia do MPC em otimizar o consumo de energia dos HBS, mantendo a temperatura e a taxa de fluxo desejadas, contribuindo para a eficiência operacional.

O trabalho de Koyfman et al. (2021) propõe um algoritmo e um programa para extrair valores de parâmetros do processo tecnológico de aquecimento do ar quente a partir de um banco de dados, com o objetivo de analisar os estados HBS durante a transição entre modos *on-gas* e *on-blast*. A estrutura de um sistema de controle automático para blocos de um HBS, baseia-se em um algoritmo capaz de ajustar os modos operacionais do bloco, permitindo aumentar a temperatura do ar quente sem a necessidade de investimentos significativos em reconstrução. Foi realizada uma análise das informações tecnológicas de um banco de dados sobre o estado do HBS durante as transições, com o intuito de ajustar os valores dos parâmetros no controle do aquecimento do ar quente. O modelo utilizado incorpora um subsistema de análise de dados, que utiliza parâmetros como consumo de gás, pressão do ar de combustão e o parâmetro do modo *on-blast*. O consumo de combustível, consumo de ar, temperatura da cúpula, temperatura máxima da fumaça, consumo de ar quente, e os tempos dos modos *on-gas* e *on-blast* foram metrificados e analisados durante os Os experimentos que foram realizados e demonstraram que a nova abordagem permite a análise em tempo real do estado do *checkerwork* dos HBS, resultando em ajustes oportunos no gráfico de parâmetros do bloco de fornos de ar quente, melhorando a eficiência do sistema de controle automático sem a necessidade de grandes investimentos em infraestrutura. Os resultados demonstraram que a aplicação do algoritmo proposto permite a estabilização da temperatura do ar quente durante o modo *on-switch*, minimizando flutuações que afetam negativamente o processo de fusão do alto-forno.

O trabalho de Navarro et al. (2022) propõe um sistema de predição da temperatura do ferro gusa como a primeira etapa de um sistema de controle maior e mais complexo com o objetivo de melhorar a eficiência da produção de aço. O estudo compara vários modelos de aprendizado de máquina (KNN, LR, EBM, LGBM, RF, SVM, XGBoost e MLP) para a predição da temperatura do ferro gusa. O estudo indica que, com um sistema de predição de temperatura, será possível planejar melhor as ações de controle antecipadamente visando estabilizar a temperatura do forno durante a produção do ferro gusa. Os experimentos usaram a base de dados fornecida por uma siderúrgica entre os anos 2017 e 2021 com 79.764 registros que foi processada e utilizada para treino e teste dos modelos. Cinco métricas foram usadas para avaliar a performance dos modelos, sendo três delas comum para esse tipo de comparação, RMSE, MAE e MAPE, além das AAR e TM. Os resultados mostram que o sistema pode prever efetivamente a temperatura do ferro gusa com $2:03 \pm 0:44$, $4:51 \pm 0:51$ e $07:40 \pm 00:58$ horas de antecedência (com erro médio absoluto de 9,55, 10,00 e 12,31, respectivamente), quando comparado com as referências

(com erro médio absoluto de 12,61, 14,91 e 19,17, respectivamente).

2.3 COMPARAÇÃO DE MODELOS

A utilização do HBS para aquecimento do ar, por meio da troca de calor, é amplamente conhecido e explorado pela indústria. Apesar disso, esse processo permanece como objeto de pesquisa contínua, visando aumentar a eficiência energética, reduzir as emissões de CO₂, os custos operacionais e ampliar a segurança operacional.

No cenário dos regeneradores, há poucos trabalhos que abordam a predição da temperatura do domo. Destacamos aqui o trabalho de Matino et al. (2019), que teve como objetivo desenvolver modelos para prever o consumo de BFG por ar HBS em siderúrgicas integradas, visando otimizar a gestão de gases residuais e melhorar a sustentabilidade do processo. Foram criados dois modelos, um modelo simples denominado *switch model* e um modelo mais complexo baseado em Redes Neurais de Estado de Eco (ESN, do inglês *Echo State Network*). O *switch model* é um modelo que não requer treinamento, operando com base na alternância regular das fases de aquecimento e sopro dos HBS. Já o modelo baseado em ESN, requer uma fase de treinamento e foi projetado para capturar o comportamento dinâmico do processo, utilizando uma camada de reservatório esparsamente conectada reproduzindo a memória intrínseca do sistema. Durante os experimentos, foi observado o consumo de BFG pelos HBS, medido em intervalos de 5 minutos para um período de previsão de até 2 horas. A precisão dos modelos foi avaliada comparando as previsões com dados reais de consumo, utilizando uma função de erro percentual. Os resultados dos experimentos mostraram que o *switch model* apresentou um erro constante na faixa de 15-16%, enquanto o modelo ESN apresentou um erro entre 8% e 12%, sendo o *switch model* mais preciso durante mudanças nas condições operacionais. No entanto, o modelo ESN mostrou-se menos estável devido à sua maior sensibilidade às variações de entrada, concluindo-se que a escolha do modelo depende das necessidades específicas do usuário, optando-se pela simplicidade com o *switch model* ou maior precisão com o modelo ESN.

Yang, Zhao e Liu (2020) aborda a predição de temperatura dos gases de exaustão de um regenerador (HBS), essencial para o controle do processo em altos-fornos. O trabalho baseia-se em redes neurais recorrentes (RNN, do inglês *Recurrent Neural Network*), com ênfase na ESN, uma variante que melhora a precisão de aprendizado em séries temporais. A Rede de Estado de Eco de Memória Profunda (DMESN, do inglês *Deep Memory Echo State Network*), aprimora a estrutura da camada oculta com a formação de estados de eco e ocultos, utilizando três portas para estabilizar o estado de eco e inicializar o estado oculto. Foram realizados pré-processamento de dados, como rejeição de outliers, tratamento de dados ausentes e cálculo de tempo de atraso. Os experimentos comparativos foram realizados entre DMESN e Leaky-ESN, utilizando as métricas RMSE, MAE e MAPE para avaliar a precisão e estabilidade dos modelos. Os resultados mostraram que a DMESN

alcançou um RMSE de 0,32, MAE de 0,24 e MAPE de 0,07, superando significativamente o Leaky-ESN. Além disso, a DMESN obteve menor tempo de treinamento, com apenas 0,09 segundos, em comparação com outros modelos. Na análise do desempenho incluiu comparações com CNN unidimensional, RNN e LSTM, onde a DMESN manteve a melhor performance em diferentes tempos de previsão (60 a 300 segundos), concluindo que a DMESN é eficaz na previsão de temperatura, superando modelos anteriores e contribuindo para a segurança e eficiência operacional do HBS.

Carmo, Mutz e Resendo (2022) abordam a predição em tempo real de gases residuais no contexto do AF, especialmente do gás Linz-Donawitz (LDG). A combinação de tarefas de classificação e regressão visa melhorar a precisão dessas previsões. Foram desenvolvidas seis estratégias para a predição da geração recuperável do LDG, sendo duas como *benchmarks* de referência, duas como modelos multi-modais multivariados utilizando GBDT-MM (do inglês *Gradient Boosting Decision Tree*) e dois modelo de rede neural LSTM (do inglês *Long Short-Term Memory*). Dados foram coletados dados de três conversores de oxigênio básico (BOF), incluindo séries temporais e dados tabulares do sistema de produção. A transformação dos dados foi realizada por meio de *pipelines*, preparando-os para a entrada nos modelos, com uma separação de 80% dos dados para treinamento e 20% reservados para testes. Nos experimentos, foram avaliados o Erro Percentual Absoluto Médio Ajustado ($MAPE_{adj}$), para medir a precisão das previsões, e o Desvio Padrão do Erro Absoluto Médio (σMAE), que mediu a variabilidade dos erros. Além disso, acurácia e precisão, foram usadas para avaliar a confiabilidade das previsões. O modelo GBDT-MM-MIX, apresentou uma redução significativa no $MAPE_{adj}$ com valores variando entre 6,97% e 19,36%, sendo classificado como o melhor modelo, enquanto o modelo LSTM-MM-MIX, apresentou-se sistematicamente o segundo melhor resultado. A pesquisa concluiu que a combinação de tarefas de classificação e regressão, utilizando modelos multi-modais multivariados, melhora a predição de LDG recuperável, a estratégia GBDT-MM-MIX estabeleceu um novo *benchmark*, devido à precisão e confiabilidade das predições, sugerindo que futuras pesquisas podem se beneficiar de otimizações adicionais e novas arquiteturas de modelos.

O trabalho de Choi et al. (2022) propõe um novo framework para prever a profundidade de geada em solos sob pavimentos utilizando oito algoritmos ML, foi utilizado como referência pois emprega técnicas de aprendizado de máquina e comparação de resultados dos modelos utilizados. A proposta visa melhorar a sustentabilidade dos pavimentos, que são vulneráveis ao congelamento e descongelamento do solo. Foram aplicados cinco algoritmos de aprendizado de máquina individuais: KNN, NN, SGD, SVM e DT, além de três técnicas de aprendizado em conjunto RF, GB e XGB. Os dados foram normalizados e divididos em 80% para treinamento e 20% para teste, com validação cruzada k-fold (k=10) para avaliar o desempenho dos modelos. As métricas de desempenho observadas foram o

coeficiente de determinação R^2 e erro quadrático médio (RMSE). Antes da otimização dos hiperparâmetros, o modelo GB apresentou um R^2 de 0.910 e um RMSE de 0.073. Após a otimização, o modelo GB continuou a mostrar o melhor desempenho com um R^2 de 0.919 e um RMSE de 0.073. Os resultados indicaram que os algoritmos de aprendizado em conjunto superaram os modelos individuais, com o modelo GB sendo o mais eficaz. As considerações finais do trabalho destacam que a otimização dos hiperparâmetros melhorou significativamente o desempenho dos modelos individuais, enquanto os modelos em conjunto já apresentavam alta performance mesmo sem otimização. Além disso, o modelo GB demonstrou ser o mais eficaz na previsão da profundidade de geada e na análise de sensibilidade dos preditores relacionados ao pavimento, sendo a espessura da superfície de asfalto como fator mais crítico.

Embora o estudo de Alshboul et al. (2024) não esteja diretamente ligado ao processo de um HBS, ele foi utilizado como referência por empregar técnicas de aprendizado de máquina para prever a resistência ao cisalhamento de vigas de concreto reforçado com fibras de aço (SFRC-SBWS), o SFRC possui propriedades de alto desempenho, que oferecem benefícios significativos de resistência e economia. Os dados foram coletados de um banco de dados atualizado com 333 espécimes de SFRC-SBWS de diversas fontes literárias. Os dados foram preparados com remoção de *outliers*, normalização e padronização, para garantir a precisão dos modelos de ML. Três técnicas de ML foram empregadas: XGBoost, GEP e LightGBM, com os dados divididos em conjuntos de treinamento (80%) e teste (20%). A análise experimental foi realizada utilizando Python 3.8, com suporte de bibliotecas como Numpy e SciPy. Nos experimentos a importância das características foi avaliada usando a correlação de Pearson e a precisão dos modelos foi utilizada métricas como Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE) e Coeficiente de Determinação (R^2). Os resultados mostraram que o LightGBM apresentou os melhores desempenhos, com (R^2) de 95,74%, RMSE de 20,04Kn, MAPE de 12,14% e MAE de 14,83. O XGBoost também demonstrou bons resultados com (R^2) de 93,27%, RMSE de 25,19Kn, MAPE de 16,23% e MAE de 19,61, que foram inferiores ao LightGBM, enquanto o GEP teve um desempenho menos preciso. O trabalho concluiu destacando que os modelos desenvolvidos superaram os existentes na literatura, demonstrando a eficácia das técnicas de ML na previsão da resistência ao cisalhamento de SFRC-SBWS.

Yang, Yan e Liu (2024) propõem um modelo de rede neural baseado em GRU chamado TSR-GRU para previsão da temperatura do domo de um HBS, abordando a complexidade dos processos industriais. O trabalho envolve a reconstrução de características em duas dimensões: tempo e espaço, atribuindo pesos diferentes para melhorar a extração de características. O modelo TSR-GRU utiliza unidades de reconstrução de tempo (TRU) e espaço (SRU) para refinar as características extraídas, reduzindo a redundância e

melhorando a representação das características. Os experimentos foram realizados com dados reais coletados de um forno de oxigênio, utilizando variáveis como fluxo de gás, fluxo de ar, temperatura da domo e emissões de gases. As métricas de avaliação incluíram o Erro Médio Absoluto (MAE), o Erro Quadrático Médio (RMSE) e o coeficiente de correlação. Os resultados mostraram que o TSR-GRU superou modelos tradicionais como CNN, LSTM, GRU, CNN-LSTM e CNN-GRU, com MAE de 0,0256, RMSE de 0,0381 e coeficiente de correlação de 0,9482. As considerações finais indicam que o TSR-GRU possui uma capacidade de extração de características mais forte e uma melhor estabilidade em previsões de longo prazo, especialmente na previsão da temperatura de domo após 60 e 80 pontos de amostragem.

A Tabela 1 apresenta um resumo dos trabalhos apresentados neste capítulo. As colunas da tabela apresentam, na sequência: a referência do trabalho; o objetivo e alvo do trabalho; os modelos propostos; as métricas utilizadas; os resultados obtidos e a classificação do trabalho. Os trabalhos citados revisam a literatura sobre a aplicação de modelos de aprendizado de máquina em previsão em ambientes industriais, além de trabalhos que focaram em métodos específicos para séries temporais, especialmente em contextos siderúrgicos. No entanto, este trabalho se diferencia ao não usar métodos tradicionais, devido ao comportamento dos dados analisados, em vez disso, adota uma abordagem diferente da encontrada na literatura ao aplicar técnicas de aprendizado de máquina em dados na previsão de temperatura do domo de regeneradores, justificando essa escolha com base nas características dos dados disponíveis. Além disso, este trabalho contribui com a literatura ao realizar uma comparação abrangente entre diversos algoritmos de aprendizado de máquina, oferecendo uma análise detalhada de suas performances na previsão da temperatura do domo de regeneradores. Essa comparação fornece *insights* sobre as diferentes abordagens para otimizar o consumo de gases na indústria siderúrgica, destacando a eficácia de modelos como LightGBM e XGBoost, que apresentaram os melhores resultados em termos de precisão.

Tabela 1 – Trabalhos correlatos

| Autor(es) | Alvo | Modelo | Métrica | Resultado | Classificação |
|---|---|---|--|---|---------------------------|
| (KWAKERNAAK; TLIJSSEN; STRUJIBOS, 1970) | - Maximização da eficiência térmica | - Algoritmo quasi-Newton | - Eficiência térmica - Período de resfriamento. | - Operação paralela escalonada é mais eficiente termicamente | - Operação Regeneradores |
| (ŞAHİN; MORARI, 2008) | - Minimizar o consumo de energia nos HBS | - Controle preditivo de Modelo (MPC) | - Temp. final do jato desvio de 0, 4°C - Temp. dos tijolos dentro da restrição de 2°C | - MPC eficaz na otimização do consumo de energia - | - Operação Regeneradores |
| (BING-YAO; HUI; YOU-JUN, 2011) | - Melhorar a previsão de temperatura final reduzindo a necessidade de ajustes manuais e reoperações | - RNN RBF Básica - RNN RBF Aprimorada - Algoritmo de <i>clustering</i> DBSCAN | - MSE=14, 95 - $\sum MSE=43.337$ | - Melhora na precisão da previsão da temperatura final no BOF, para cenários industriais | - Predição de Temperatura |
| (TANG; LIU, 2015) | - Melhorar a precisão estática e as características dinâmicas do controle de temperatura | - Modelo Matemático - Rede Neural de três camadas | - Média da variância do erro | - Controle preditivo baseado em redes neurais robusto | - Predição de Temperatura |
| (MATINO et al., 2019) | - Predição de consumo de BFG em HBS - Otimização de gestão de gases | - Modelo simples switch model - Modelo em Redes Neurais de Estado de Eco (ESN) | - Erro entre 8% e 12% | - ESN o mais adequado | - Comparação de modelos |
| (YANG; ZHAO; LIU, 2020) | - Predição de gases de exaustão | - DMESN+ Leaky-ESN - CNN unidimensional+RNN+LSTM | - RMSE=0, 32 - MAE=0, 24 - MAPE=0, 07 | - DMESN é eficaz na predição de temperatura | - Comparação de Modelo |
| (KOYFMAN et al., 2021) | - Extração de parâmetros do <i>On-blast</i> - Melhorar a eficiência do <i>On-blast</i> | - Algoritmo - Programa de extração de dados - MARS | - N/A | - Estabilizou a temperatura do ar no modo <i>On-switch</i> | - Operação Regeneradores |
| (KACUR et al., 2022) | - Predição da temperatura e da concentração de carbono no processo de fabricação de aço em BOF | - SVR - KNN - RNN - RF | - MAE - RMSE - PI | - MARS melhor desempenho na predição da temperatura - KNN melhor predição de concentração de carbono | - Predição de Temperatura |
| (CARMO; MUTZ; RESENDO, 2022) | - Predição de gases residuais | - Dois <i>benchmarks</i> de referência - Dois modelos GBDT-MM - Dois modelos rede neural LSTM | - MAPE _{vari} entre 6,97% e 19,36% - σ MAE - Acurácia | - GBDT-MM-MIX | - Comparação de modelos |
| (CHOI et al., 2022) | - Previsão da profundidade de geada em solos | - KNN -NN -SGD -SVM -DT -RF -GB -XGB | - RMSE=0, 073 R ² = 0, 919 | - GB mais eficaz | - Comparação de modelos |
| (ALSHBOUL et al., 2024) | - Predição de resistência ao cisalhamento de vigas de concreto | - XGBoost - GEP - LightGBM | - RMSE= 20,04kN - MAPE= 12,13% - MAE= 14,83% - R ² = 95,74% | - LightGBM - Melhor -XGBoost - Segundo | - Comparação de Modelos |
| (YANG; YAN; LIU, 2024) | - Previsão de temperatura de domo com GRU | - TSR-GRU -CNN -LSTM -GRU -CNN e LSTM - CNN e GRU | - RMSE=0, 0381 R ² = 0, 9482 | - TSR-GRU tem melhor predição a longo prazo | - Comparação de modelos |

Fonte: Próprio autor.

3 MATERIAIS E MÉTODOS

Neste capítulo, são descritos os procedimentos para coleta e pré-processamento da base de dados utilizada para treinar e avaliar os modelos. A seguir é apresentada uma descrição dos modelos e das métricas usadas para comparação dos resultados.

3.1 COLETA DE DADOS

Durante o preparo dos AFs, centenas de dados são coletados, descrevendo o estado desses equipamentos e dos demais que compõem uma planta siderúrgica integrada. Para este estudo, foram reunidos dados reais de quatro HBS, utilizados para gerar ar de alta temperatura para três AFs em uma empresa de siderurgia. Os dados analisados nesta pesquisa foram registrados a cada 36 minutos, aproximadamente. A Tabela 2 descreve as oito tabelas de banco de dados, e a quantidade de amostras registradas pelos sensores dos quatro HBS que atendem aos três AF durante os anos de 2018 a 2023. Ao todo, 1.904.491 amostras foram coletadas.

A base de dados totaliza, conforme a Tabela 2, 178 atributos. Esses atributos resultam de sensores instalados nos HBS, sensores de informações suplementares, ações de controle dos operadores e resultados de cálculos matemáticos utilizados para a avaliação do estado durante os processos de *on-blast* e *on-gas*. Entre os atributos, estão as chaves primárias e estrangeiras das tabelas, que permitem a identificação exclusiva das amostras, facilitando o relacionamento entre as tabelas. Para este estudo, focamos nas amostras do ano de 2023 dos três AFs, totalizando um conjunto com 155.103 amostras.

Tabela 2 – Tabelas de dados coletadas dos Regeneradores e seus AFs.

| Tabela | AF1 | AF2 | AF3 | Descrição |
|-------------------------|---------------|----------------|------------------|--|
| hs_ciclos | 50.738 | 53.827 | 40.143 | Registros das datas dos ciclos de cada regenerador |
| hs_combustoes | 1.966 | 92.090 | 72.118 | Dados dos sensores de combustão de cada ciclo |
| hs_guiia_estagios | 4.256 | 26.277 | 180.288 | Dados dos sensores em cada período de combustão |
| hs_indices_operacionais | 16.681 | 22.897 | 22.435 | Dados dos índices operacionais por datas |
| hs_medias_diarias | 2.893 | 2.804 | 3.198 | Cálculo da media diária dos indicadores |
| hs_sopros | 1.966 | 92.088 | 72.111 | Dados dos sensores durante do sopro do ar frio de cada ciclo |
| hs_temperaturas_grelhas | 1.966 | 92.088 | 360.555 | Dados da temperatura das grelhas durante o sopro |
| hs_temperaturas_silicas | 1.966 | 92.088 | 144.222 | Dados da temperatura das sílicas durante o sopro |
| Total | 82.442 | 710.656 | 1.111.403 | |

Fonte: Próprio autor.

3.2 SELEÇÃO DE ATRIBUTOS

O estudo foi conduzido utilizando as amostras descritas, que, inicialmente, apresentavam 178 atributos de controle, juntamente com informações suplementares relacionadas às medições durante os processos *on-blast* e *on-gas*. Neste estudo, o foco foi no conjunto de amostras do HBS de número um (HBS-1), totalizando 5.684 amostras coletadas ao longo do ano de 2023.

A base de dados original é composta por 178 atributos. Com o objetivo de identificar os atributos mais relevantes para o estudo, foram realizadas entrevistas com especialistas do processo e, em seguida, aplicou-se um método estatístico para a redução de dimensionalidade. Inicialmente, foram removidos os atributos do tipo chave primária das tabelas (ID), bem como atributos de registro de datas e variáveis de segunda ordem, como médias de outros atributos. Como resultado parcial, foram selecionados 97 dos 178 atributos originais. Posteriormente, aplicou-se a Análise de Componentes Principais (PCA, *Principal Component Analysis*) (XIE; KRISHNAN et al., 2014), a fim de determinar a correlação entre a variável alvo e as demais variáveis da base de dados, conforme demonstra a Tabela 3. Segundo os resultados do PCA, os três primeiros atributos são capazes de representar quase 100% da variância total do atributo alvo (TEMPERATURA_DOMO_INICIAL), como observado na coluna % Variância nas 3 primeiras linhas. Na tabela, os atributos de entrada estão listados, e as colunas descrevem o percentual de correlação com a variável alvo (peso) e a porcentagem de variância total do componente presente.

3.3 PRÉ-PROCESSAMENTO DOS DADOS

Além da preparação dos atributos mencionados, foi necessário realizar a limpeza e o pré-processamento dos dados, conforme os procedimentos detalhados a seguir. O primeiro procedimento foi remover registros duplicados, a fim de garantir a integridade e confiabilidade dos dados utilizados no conjunto de entrada.

O conjunto de dados utilizado neste estudo apresentou valores ausentes em alguns dos atributos. Para solucionar esse problema, aplicamos a técnica de imputação pela média móvel, utilizando uma janela fixa que considera as 24 amostras mais recentes em relação ao valor ausente. Esse método abrange um período de aproximadamente 14 horas de operação, contemplando no mínimo um ciclo de *on-gas* e um ciclo de *on-blast*. A imputação pela média móvel com uma janela de 24 amostras é eficaz para capturar a tendência e a sazonalidade dos dados, evitando que valores muito distantes no tempo influenciem indevidamente o preenchimento dos valores faltantes. No entanto, é importante destacar que essa técnica pode suavizar ou alterar os padrões originais dos dados, o que pode afetar a qualidade dos modelos caso haja um período prolongado de valores ausentes. Porém, esse não foi o caso no conjunto de dados.

Tabela 3 – Resultado do PCA - peso e % variância das variáveis.

| Atributos | Peso | % Variância |
|----------------------------------|-----------|-------------|
| 1 CALOR_GERADO_ACUMULADO | 0,976241 | 82.6951 |
| 2 PCI_GAS_MISTO_ACUMULADO | 0,198358 | 16.8024 |
| 3 VAZAO_AR_ACUMULADO | 0,062290 | 5.2764 |
| 4 CALOR_NECESSARIO_UNITARIO | 0,019691 | 1,6679 |
| 5 PRESSAO_GAS_MISTO_MEDIA | 0,013285 | 1,1253 |
| 6 TEMPERATURA_CHAMA_INICIO | 0,008206 | 0,6951 |
| 7 TEMPERATURA_CHAMA_FIM | 0,007772 | 0,6583 |
| 8 PRESSAO_AR_AQUECIDO_MEDIA | 0,004398 | 0,3725 |
| 9 TEMPERATURA_AR_INICIO | 0,004101 | 0,3473 |
| 10 TEMPERATURA_CHAMINE_FIM | 0,003610 | 0,3057 |
| 11 TEMPERATURA_CHAMINE_INICIO | 0,003536 | 0,2995 |
| 12 TEMPERATURA_GAS_MISTO_INICIO | 0,002330 | 0,1973 |
| 13 TEMPERATURA_GAS_MISTO_FIM | 0,002218 | 0,1878 |
| 14 TEMPERATURA_ANTERIOR_DOMO | 0,000088 | 0,0074 |
| 15 ESTAGIO_ATUAL | 0,000012 | 0,0009 |
| 16 CALOR_CHAMINE_ACUMULADO | 0,000004 | 0,0003 |
| 17 ENRIQUECIMENTO_OXIGENIO_FIM | -0,000016 | -0,001356 |
| 18 O2_CHAMINE_MEDIA | -0,000023 | -0,001925 |
| 18 POTENCIA_CHAMA_FIM | -0,001978 | -0,167526 |
| 20 PUREZA_EO2_COMB_MEDIA | -0,002026 | -0,171630 |
| 21 POTENCIA_CHAMA_INICIO | -0,002198 | -0,186167 |
| 22 VAZAO_CHAMINE_ACUMULADO | -0,014982 | -1,269147 |
| 23 PCI_GAS_MISTO_SEM_EO2_FIM | -0,024860 | -2,105994 |
| 24 VAZAO_OXIGENIO_COMB_ACUMULADO | -0,026119 | -2,212660 |
| 25 VAZAO_GAS_MISTO_ACUMULADO | -0,026221 | -2,221325 |
| 26 TEMPERATURA_DOMO_COMB_INICIO | -0,027188 | -2,303231 |

Fonte: Próprio autor.

Após o tratamento dos dados ausentes, prosseguimos com a filtragem dos dados utilizando uma técnica baseada no IQR (*Interquartile Range*), conforme a Equação 1. Essa estratégia visa tornar os dados mais comparáveis, reduzindo a influência de valores extremos e promovendo uma distribuição mais homogênea. Os limites inferior e superior foram estabelecidos com base nos quartis do conjunto de dados, conforme a Equação 2. Valores fora desses limites, que representaram 1,3% da base dados, foram considerados como *outliers*. Para garantir a continuidade temporal, os *outliers* foram transformados em valores ausentes e em seguida aplicamos novamente o tratamento de imputação por média móvel.

$$X_{norm} = \frac{x - Q_1}{(IQR)}; \quad IQR = (Q_3 - Q_1) \quad (1)$$

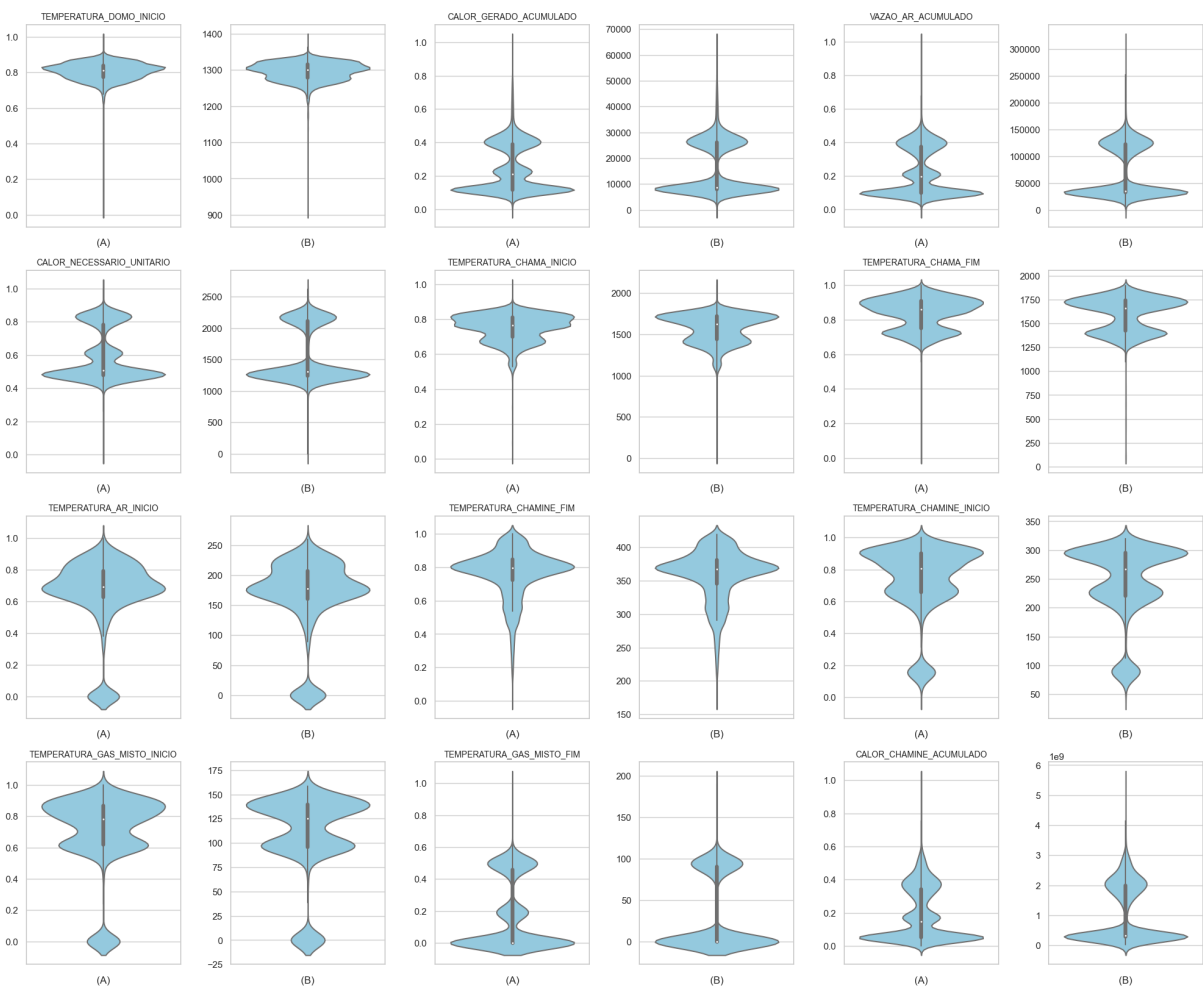
$$\text{Inferior} = Q_1 - 1,5IQR;$$

$$\text{Superior} = Q_3 + 1,5IQR$$

(2)

Para visualizar o impacto do tratamento de valores ausentes e *outliers* nas variáveis do conjunto de dados, utilizamos gráficos do tipo *ViolinPlot*, conforme apresentado na Figura 3. A Figura 3 apresenta as variáveis de maior peso de acordo com a Tabela 3, a primeira imagem (A) da Figura 3 apresenta os valores antes do processamento, enquanto a segunda imagem (B) da Figura 3 são os dados após o tratamento de valores ausentes e *outliers*. Ao analisar a Figura 3, observamos que os dados apresentam-se mais concentrados na figura (B), com uma menor dispersão, evidenciado pela redução da altura do “violino”. Além disso, a simetria dos violinos indica que a distribuição dos dados se tornou mais simétrica após o tratamento.

Figura 3 – *ViolinPlot* para comparar os dados antes e depois do pré-processamento.



Fonte: Fonte: Próprio autor.

No passo seguinte, realizamos a normalização das variáveis com o objetivo de transformar todas as variáveis do conjunto na mesma ordem de grandeza. Foi utilizada a técnica

de *Min-Max Scaling*, apresentada na Equação 3. A normalização foi feita para todas as variáveis.

$$X_{normal} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (3)$$

Por fim, foi utilizado o método de validação cruzada *k-fold* de 5, para divide o conjunto de dados em 5 subconjuntos mutualmente exclusivo para treinamento e teste, conforme descrito por Salazar et al. (2022).

A validação cruzada *k-fold* é uma alternativa à abordagem do conjunto de validação. Esta técnica separa os dados em *k* dobras de aproximadamente o mesmo tamanho. Em seguida, a validação cruzada é realizada *k* vezes, onde o subconjunto *k* específico é aplicado como conjunto de teste, enquanto as *k - 1* dobras restantes são utilizadas para treinar o modelo. Finalmente, o procedimento é repetido *k* vezes, alternando a dobra do teste, e eventualmente estimando o erro de validação cruzada *k - fold* como o erro médio de predição sobre *k* folds, conforme Hastie et al. (2009) e James et al. (2013), neste processo foi utilizado o *k-fold* com *split* de 5.

Neste estudo, a divisão foi feita com 80% das amostras destinadas ao conjunto de treinamento. Os 20% restantes ao conjunto de teste.

3.4 MODELOS INVETIGADOS

3.4.1 K-Nearest Neighbors (KNN)

O *K-Nearest Neighbors* (KNN) é um algoritmo de aprendizado supervisionado usado para classificação ou para regressão, de acordo com Zhang et al. (2017), o KNN classifica novos dados com base nos exemplos mais próximos dado a suas características, denominado método de vizinhos mais próximos. O KNN realiza tarefas de classificação calculando inicialmente a distância entre a amostra de teste e todas as amostras de treinamento para obter seus vizinhos mais próximos e, em seguida, conduzindo a classificação de KNN Com sua implementação simples e desempenho de classificação significativo, o método KNN é muito popular em mineração de dados e estatística e, portanto, foi votado como um dos dez principais algoritmos de mineração de dados segundo Zhang (2012). Atualmente o KNN tem sido aplicado em diversas áreas, como, Visão Computacional para reconhecimento de padrões e classificação de imagens e dados de visão rotulados (ZHU et al., 2020), classificação de imagens que é uma tarefa importante no campo do aprendizado de máquina e processamento de imagens(DANG et al., 2018), sugestão de produtos baseados em padrões de preferências de usuários semelhantes (SUBRAMANIYASWAMY; LOGESH, 2017).

3.4.1.1 Armazenamento dos Dados

O armazenamento dos dados de treinamento, não constrói um modelo preditivo explícito durante a fase de treinamento, ou seja, os dados são armazenados e utilizados diretamente na fase de classificação e em sua forma básica, em que para um registro de dados t seja classificado, seus k vizinhos mais próximos são recuperados, formando uma vizinhança de t . A observação majoritária entre os registros de dados na vizinhança é geralmente usada para decidir a classificação para t . Entretanto, é necessário escolher um valor apropriado para k , sendo o sucesso da classificação dependente desse valor. O KNN é categorizado como um algoritmo “*lazy*”, uma vez que ele precisa de $O(n^2)$ para classificar uma nova instância, embora seja de fato menos dependente de k e seja capaz de atingir um desempenho de classificação próximo ao do melhor k (GUO et al., 2003).

3.4.1.2 Escolha do Parâmetro k e Identificação dos k Vizinhos Mais Próximos

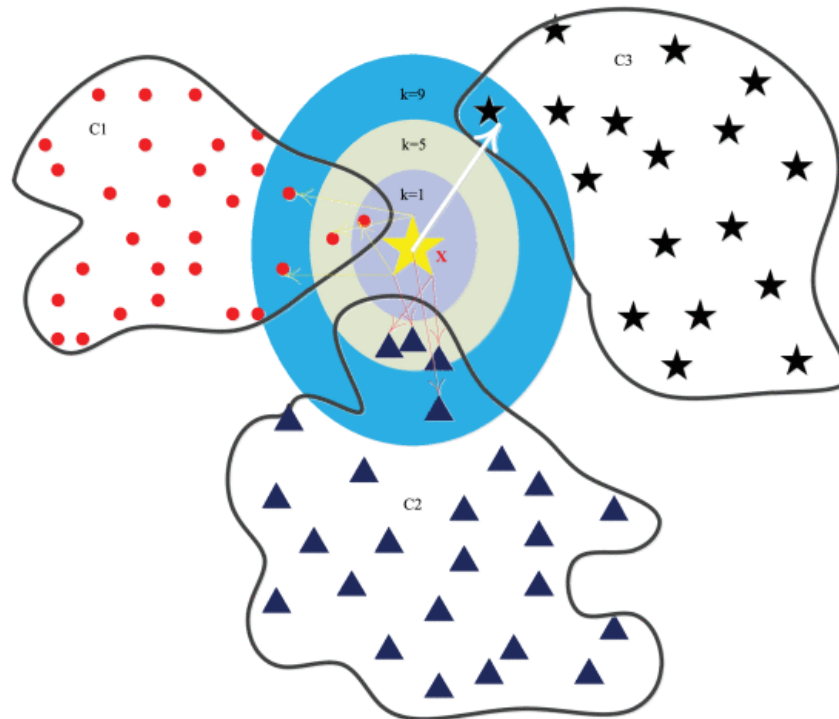
A identificação dos k vizinhos mais próximos, se dá após calcular as distâncias entre os pontos, o algoritmo seleciona os k pontos mais próximos do espaço de treinamento, configurando a base para a classificação subsequente. Os vizinhos k -mais próximos (KNN), é baseada na ideia de que os padrões mais próximos de um padrão alvo x , em que se busca rotulá-lo e fornecer informações úteis sobre o rótulo, é atribuído o rótulo de classe da maioria dos padrões k -mais próximos no espaço de dados. Portanto é necessário definir uma medida de similaridade no espaço de dados (KRAMER et al., 2013).

A escolha do valor de k é crítica para o desempenho do KNN. Um k pequeno pode resultar em um modelo excessivamente sensível a ruídos, levando ao *overfitting*, que é como o fenômeno que ocorre quando um modelo de aprendizado de máquina se ajusta excessivamente aos dados de treinamento, memorizando peculiaridades e ruídos em vez de encontrar uma regra preditiva geral. Isso acontece quando se trabalha arduamente para encontrar o melhor ajuste para os dados de treinamento, o que pode resultar em uma performance ruim ao aplicar o modelo em novos dados (DIETTERICH, 1995). Enquanto um k elevado pode suavizar a fronteira de decisão, resultando em *underfitting* que significa que o treinamento é insuficiente e a precisão do aprendizado é baixa não podem funcionando bem para dados de teste, levando a uma fraca adaptabilidade (ZHANG; ZHANG; JIANG, 2019).

A Figura 4 ilustra como a escolha de k influencia a formação das regiões de decisão no espaço de características. Com um k pequeno ($k = 1$), o círculo interno, em azul demonstra que as regiões de decisão são mais complexas e detalhadas, refletindo a variabilidade local dos dados. À medida que k aumenta ($k = 5$), círculo verde, essas regiões se tornam um pouco mais amplas e porém menos detalhadas, refletindo uma visão com mais vizinhos para classificação, por fim, para valores ainda maiores de k ($k = 9$), círculo em azul claro,

essas regiões se tornam mais amplas e menos detalhadas, refletindo uma visão mais global dos dados.

Figura 4 – KNN classificação.



Fonte: Fonte:(XING; BEI, 2019).

Para estimar dados específico, o KNN assume ser constante em uma vizinhança ao redor de um ponto x . Se v for o volume da vizinhança e de observações caírem nessa região, a estimativa de densidade do vizinho mais próximo em x é dada pelo ponto x . Se a mesma vizinhança for usada para estimativa de todas as densidades populacionais, substituindo essas estimativas na regra de Bayes, obtém-se o classificador de vizinho mais próximo usual, que essencialmente classifica uma observação para a classe que tem o número máximo de representantes naquela vizinhança. Normalmente, uma bola fechada de raio r é considerada como esta vizinhança, representando a distância entre x e seu k -ésimo ponto de dados mais próximo na amostra de treinamento, e a regra de classificação resultante é conhecida como regra do k -vizinho mais próximo (GHOSH, 2006).

A alta dimensionalidade reduz a eficácia do KNN, tornando irrelevante a diferenciação entre distâncias em alta dimensionalidade. Técnicas de redução de dimensionalidade, como o PCA e t-SNE, são utilizadas para superar essa limitação (TEHRANI et al., 2023).

3.4.1.3 Classificação e Regressão

Durante a classificação, O KNN busca k pontos de dados no conjunto de treinamento que são mais relevantes para um dado de teste os k vizinhos mais próximos. A classificação é realizada com base na classe que ocorre com mais frequência entre esses vizinhos, conhecida

como regra da maioria. Este método é semelhante à regra Bayesiana, em que a classe predominante entre os vizinhos determina a etiqueta do dado de teste (ZHANG, 2021).

A Figura 4 uma representação visual de como o KNN classifica novos dados com base na proximidade e na densidade de classes no espaço de características, ilustrando o método em capturar a estrutura local dos dados para a classificação. No diagrama, diferentes valores de k são representados para mostrar como a variação de k pode afetar a decisão de classificação. Quando k é pequeno ($k = 1$), considera apenas os vizinhos mais próximos, fazendo com que o algoritmo tenda a ser mais sensível ao ruído nos dados, podendo levá-lo a uma classificação errônea, principalmente se este vizinho se tratar de um *outliers*. Por outro lado, um valor de k muito grande ($k = 9$) pode suavizar demais a fronteira de decisão, levando a uma classificação menos precisa, pois considera muitos vizinhos que podem pertencer a diferentes classes.

O algoritmo de classificação KNN, destacando o processo de decisão baseado nos vizinhos mais próximos. No contexto do KNN, cada ponto de dados no conjunto de teste é classificado com base na classe predominante entre seus k vizinhos mais próximos no conjunto de treinamento, expandindo a regra do vizinho mais próximo para considerar múltiplos vizinhos (k), permitindo que o KNN utilize mais informações para a tomada de decisão. Isso é visualizado na figura através de um ponto de teste cercado por vários pontos de dados de diferentes classes, onde a classe do ponto de teste é determinada pela maioria dos vizinhos mais próximos, fazendo uma representação visual de como o KNN classifica novos dados com base na proximidade e na densidade de classes no espaço de características, ilustrando como o método capturar a estrutura local dos dados para a classificação.

A regressão KNN estima a função tomando uma média local do conjunto de dados de treinamento. A localidade é definida em termos das k amostras mais próximas da amostra de estimativa. Como o desempenho do algoritmo KNN depende fortemente do parâmetro k , encontrar os melhores valores de k é essencial. Um valor grande de k diminui o efeito do ruído e minimiza as perdas de previsão. No entanto, um valor pequeno de k permite uma implementação simples e consultas eficientes (FARAHNAKIAN et al., 2013).

$$\hat{y} = \frac{1}{k} \sum_{i=1}^k y_{(i)} \quad (4)$$

sendo:

\hat{y} - Representa o valor médio (ou estimado) calculado.

k - É o número total de elementos no conjunto.

$\sum_{i=1}^k y_{(i)}$ - É a soma de todos os valores $y_{(i)}$ no conjunto, onde i varia de 1 até k .

A Equação 4 representa matematicamente a previsão da regressão KNN, onde k é o número de vizinhos mais próximos considerados, e $y_{(i)}$ são os valores de saída dos k vizinhos mais próximos.

3.4.1.4 Métricas de desempenho

A análise de desempenho do algoritmo KNN pode ser avaliada por meio de uma gama diversa de métricas, que abrange a *accuracy*, matriz de confusão, *recall*, F1-score, curva ROC essas métricas são essenciais para entender a eficácia do algoritmo em diferentes aplicações.

$$Accuracy = \frac{\sum_{i=1}^n N_{ii}}{\sum_{i=1}^n \sum_{j=1}^n N_{ij}} \quad (5)$$

sendo:

N_{ii} - Os elementos da diagonal principal da matriz de confusão.

$\sum_{i=1}^n N_{ii}$ - Soma de todas as previsões corretas (diagonal principal da matriz de confusão).

$\sum_{i=1}^n \sum_{j=1}^n N_{ij}$ - Soma de todos os elementos da matriz de confusão (corretas e incorretas).

A *accuracy* do KNN pode variar com base na escolha de k e no conjunto de dados, esta medida é determinada pela porcentagem de instâncias que são categorizadas corretamente, é uma estatística crucial para avaliar a eficácia dos modelos (GENG, 2024). A Equação 5 define a *accuracy* como a métrica que avalia a frequência de acertos de um modelo de predição, expressa pela porcentagem de previsões corretas em relação ao total de previsões realizadas feitas (ALIFIA et al., 2023).

Figura 5 – Exemplo de Matriz de Confusão.

| | | Predicted | | | |
|--------|----------|-----------|------------------|----------|--|
| | | A_1 | ... A_j ... | A_n | |
| Actual | A_1 | N_{11} | N_{1j} | N_{1n} | |
| | \vdots | | \vdots | | |
| | A_j | N_{i1} | ... N_{ij} ... | N_{in} | |
| | \vdots | | \vdots | | |
| | A_n | N_{n1} | N_{nj} | N_{nn} | |

Fonte: Fonte:(DENG et al., 2016).

A matriz de confusão é uma ferramenta visual e avaliação de modelos de classificação, fornecendo uma visão da predições corretas e incorretas para cada classe, possibilitando detalhar as predições verdadeiros positivos, falsos positivos, verdadeiros negativos e falsos

negativos. A matriz é composta por dois eixos: o eixo y representa as classes reais dos exemplos e o eixo x representa as classes previstas pelo modelo (DENG et al., 2016). A Figura 5 apresenta a forma básica da matriz de confusão para uma tarefa de classificação multiclasse, com as classes A_1, A_2, \dots, A_n . Na matriz de confusão, N_{ij} representa o número de amostras que realmente pertencem à classe A_i , mas que foram classificadas como classe A_j .

$$Precision_i = \frac{N_{ii}}{\sum_{k=1}^n K_{ki}} \quad (6)$$

sendo:

N_{ii} - Os elementos da diagonal principal da matriz de confusão.

$\sum_{k=1}^n K_{ki}$ - Soma de todas as previsões feitas (corretas e incorretas) para a classe i .

O *Precision* é uma métrica que mede o quão bem o modelo consegue identificar instâncias positivas, objetivando reduzir os falsos positivos. A Equação 6 descreve matematicamente o *Precision*.

$$Recall_i = \frac{N_{ii}}{\sum_{k=1}^n K_{ik}} \quad (7)$$

sendo:

N_{ii} - Os elementos da diagonal principal da matriz de confusão.

$\sum_{k=1}^n K_{ik}$ - Soma de todos os exemplos reais da classe i (incluindo os que foram classificados corretamente e incorretamente).

O *Recall* mede a capacidade do modelo de previsão de selecionar instâncias de uma determinada classe a partir de um conjunto de dados, a Equação 7 define sua fórmula matemática (DENG et al., 2016).

$$F - Score_i = \frac{2 \cdot Precision_i \cdot Recall_i}{Precision_i + Recall_i} \quad (8)$$

sendo:

$Precision_i$ - A proporção de previsões corretas para a classe i (evita falsos positivos).

$Recall_i$ - A proporção de exemplos reais da classe i que foram corretamente identificados (evita falsos negativos).

O *F1-Score* demonstra uma pontuação para avaliar o desempenho de um modelo, equilibrando precisão e *Recall*, sendo o *F-score* tradicional a média harmônica da *Precision* e do *Recall*, a Equação 8 demonstra a formula matemática do *F-score* (GENG, 2024) (DENG et al., 2016).

$$f(x) = \sum_{m=0}^M f_m(x) \quad (9)$$

sendo:

y - O valor real ou esperado.

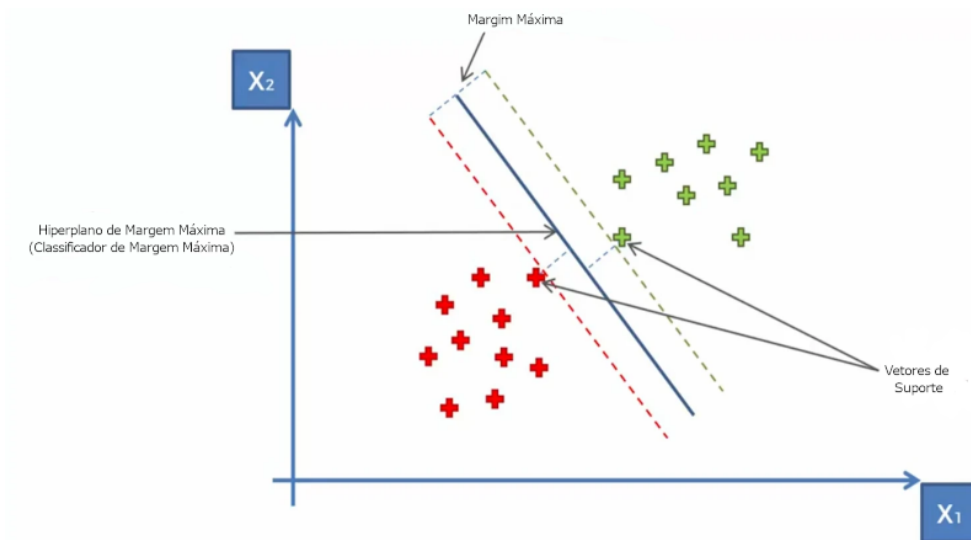
$f(x)$ - A saída ou previsão do modelo para a entrada x .

3.4.2 Support Vector Machines (SVM)

As *Support Vector Machines* (SVM) são algoritmos de aprendizado de máquina populares devido à robustez e eficácia na classificação de conjuntos de dados. O SVM opera com baixa exigência de memória, uma vez que se concentram em subconjuntos do conjunto de treinamento. O principal objetivo do SVM é estabelecer a melhor linha ou hiperplano de decisão para separar pontos em um espaço bidimensional, categorizando-os de forma clara entre as possíveis classes, como ilustrado na Figura 6. Os elementos importantes para o funcionamento são os vetores de suporte, que são pontos de dados mais próximos do plano de separação e influenciam diretamente na definição da linha divisória. Outro conceito essencial é a margem, que se refere à distância entre os pontos mais próximos das diferentes classes e o hiperplano, sendo fundamental para maximizar a separação e garantir a precisão do modelo (AZIZ et al., 2022).

Existem dois tipos principais de classificadores SVM: os lineares e não lineares. Os classificadores lineares são projetados para separar pontos de dados usando uma linha reta (ou um hiperplano em dimensões maiores). Nesse tipo, o limite de decisão é estabelecido de forma que uma classe esteja em um lado do limite e a outra classe no lado oposto, como demonstrado na Figura 6. Nessa figura, é possível observar que os itens da classe verde estão do lado oposto da classe vermelha. Isso é obtido definindo uma margem que maximize a distância entre os pontos mais próximos das duas classes, conhecidos como vetores de suporte. O SVM linear é eficaz quando os dados são linearmente separáveis, ou seja, quando uma linha reta pode dividir as classes sem sobreposição. Por outro lado, os classificadores SVM não lineares são usados quando os dados não podem ser separados por uma linha reta, como apresentado na Figura 7. Nessa figura, amostras da classe vermelha e verde, embora agrupados, não podem ser separados por uma linha reta. Nesse caso, a flexibilidade do SVM entra em ação por meio do uso das funções de kernel. O kernel é uma função de similaridade associada ao algoritmo de aprendizado de máquina quantifica

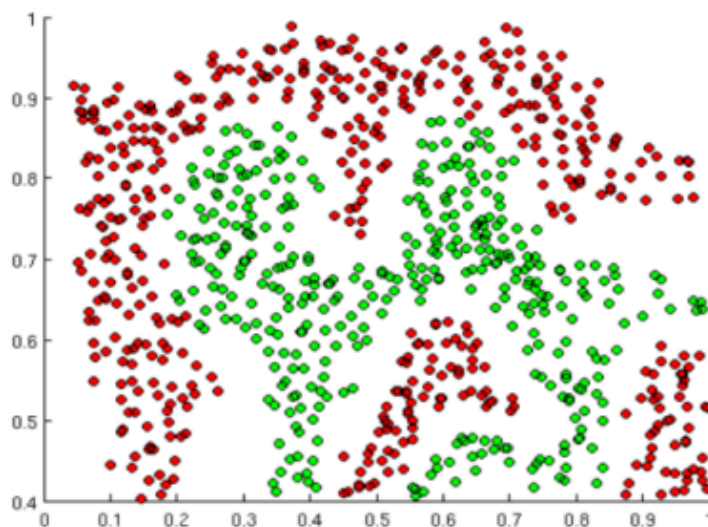
Figura 6 – Limite de decisão na separação do espaço e classe.



Fonte: Fonte:(AZIZ et al., 2022) e adaptado pelo autor.

a similaridade entre duas entradas (amostras). Inicialmente, o SVM foi projetado para classificações lineares, mas os avanços permitiram lidar com classificações não lineares ao transformar o espaço de entrada em um espaço de maior dimensão usando kernels. Esses kernels podem ser de diferentes tipo, como: Polinomial Kernel, que permite limites de decisão curvos ao considerar as relações polinomiais entre as características. Sigmoid Kernel: Semelhante à função de ativação usada em redes neurais, pode criar hiperplanos de separação complexos. Kernel da Função de Base Radial (RBF): Um dos kernels mais populares, capaz de lidar com casos em que a relação entre rótulos e atributos de classe não é linear (GHOSH; DASGUPTA; SWETAPADMA, 2019).

Figura 7 – SVM- Classificação Não Linear.



Fonte: Fonte:(AZIZ et al., 2022).

3.4.3 Random Forests (RF)

O método *Random Forests* (RF), que utiliza decisões aleatórias, foi inicialmente proposto por Heath, Kasif e Salzberg (1993) em 1993. O algoritmo RF é derivado da técnica de *Bagging*, ou *Bootstrap Aggregation*, introduzida por Breiman (1996) em 1996. Segundo o estudo de Lee, Ullah e Wang (2020), a implementação do *Bagging* pode reduzir significativamente a variância associada a um modelo preditivo em comparação com a aplicação do modelo na amostra inicial de dados, oferecendo, assim um mecanismo para aumentar a confiabilidade das previsões.

O RF é um modelo de aprendizado de máquina amplamente utilizado para classificação e predição, que requer uma grande quantidade de dados de alta qualidade para o seu treinamento. As *Random Forests* são criadas por meio da geração de múltiplas árvores de decisão. Isso é alcançado através da coleta de amostras aleatórias de dados usando o método *Bootstrap*, além da seleção aleatória de características de entrada. Cada árvore de decisão dentro do RF é considerada uma árvore simples.

Uma das vantagens das florestas aleatórias é a alta precisão que elas proporcionam em comparação com outras abordagens, como *Bagging* e *Boosting*, conforme destacado por Salman, Kalakech e Steiti (2024). Essa precisão é alcançada por de sua capacidade de lidar com a variabilidade e a incerteza dos dados de entrada, resultando em previsões mais confiáveis e robustas.

3.4.3.1 Árvores de Decisão

A árvore de decisão é um modelo de aprendizado de máquina supervisionado utilizado para classificação e regressão, que representa decisões e suas possíveis consequências em uma estrutura de árvore. Cada nó interno dentro da árvore representa uma “decisão” baseada em um atributo, enquanto cada folha indica um resultado ou classe resultante. O modelo é construído particionando recursivamente os dados em subconjuntos baseados em avaliações de atributos, aprimorando assim a interpretação e a visualização dos processos de tomada de decisão. As árvores de decisão são estimadas por sua simplicidade e capacidade de acomodar dados categóricos e contínuos (DONG et al., 2020).

No entanto, as árvores de decisão apresentam algumas desvantagens tais como: propensão ao *overfitting*, especialmente em conjuntos de dados complexos, uma vez que a árvore pode se ajustar excessivamente aos dados de treinamento, resultando em baixa generalização para novos dados (SALMAN; KALAKECH; STEITI, 2024). Além disso, a instabilidade devido a pequenas variações nos dados de entrada pode resultar em árvores de decisão muito diferentes, tornando o modelo instável. Há também a tendência de criar árvores muito profundas, o que dificulta a interpretação e a visualização do modelo (DONG et al., 2020).

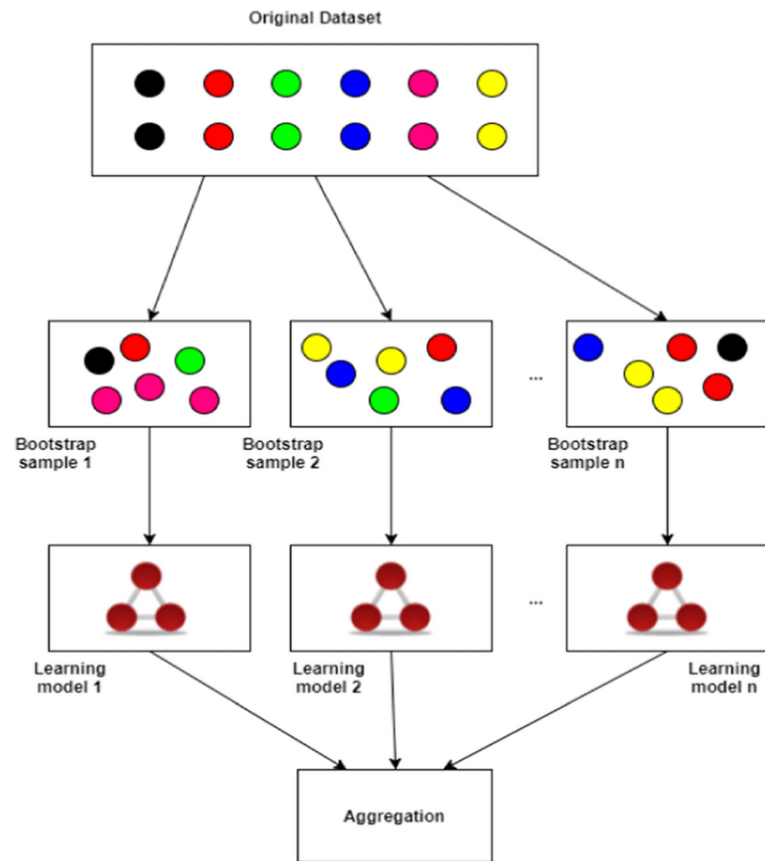
O RF agrega as previsões de várias árvores de decisão, tendendo a fornecer previsões mais robustas e precisas do que uma única árvore de decisão, tornando-o especialmente útil em problemas complexos (XUE et al., 2021). O RF desenvolve uma série de modelos de árvores de decisão a partir de dois pontos de vista distintos: a dimensão das amostras e a dimensionalidade dos atributos. Conseqüentemente, visa reduzir os desafios associados às árvores de decisão, combinando os resultados da votação de várias árvores de decisão (DONG et al., 2020).

3.4.3.2 Bagging ou Bootstrap aggregation

Segundo Salman, Kalakech e Steiti (2024), o método *Bagging*, ou *Bootstrap Aggregation*, é uma técnica de *ensemble learning* que gera subconjuntos de amostras através de amostragem aleatória do conjunto de dados de treinamento. Esses subconjuntos são então utilizados para treinar modelos básicos de forma paralela. O objetivo do *Bagging* é integrar os resultados desses modelos básicos para melhorar a precisão e a robustez do modelo final ajudando a reduzir o *overfitting*, especialmente em modelos como árvores de decisão, ao combinar os resultados de múltiplos modelos treinados em diferentes subconjuntos de dados.

A Figura 8 ilustra o funcionamento do modelo *Bagging*. O primeiro passo (*Bootstrap sample*) consiste em gerar n novas amostras de treinamento, cada uma com o mesmo tamanho do conjunto original, formando subconjuntos de dados de treino. No passo seguinte (*Learning model*), cada árvore é treinada individualmente com os novos subconjuntos de amostras de treinamento criados na etapa anterior. Na etapa final, sendo um algoritmo de regressão ou agregação, é calculada a média das previsões, obtendo-se uma previsão final, enquanto em um algoritmo de classificação é utilizada a maioria dos votos.

A ideia central, ao combinar múltiplas previsões, é que o *bagging* reduz a variância do erro de previsão no modelo final, que é treinado com diferentes conjuntos de dados, resultando em previsões mais assertivas. O método é particularmente eficaz quando pequenas mudanças nos dados de treinamento podem levar a grandes variações no modelo, como em árvores de decisão e redes neurais. Sua vantagem é o fato de que modelos individuais treinados com base em diferentes dados de treinamento têm seus erros reduzidos durante a média. No entanto, para métodos estáveis, como k -vizinhos mais próximos, o *bagging* pode não oferecer melhorias significativas. Além disso, ele pode tornar o modelo final menos interpretável, especialmente quando aplicado a árvores de decisão, já que a estrutura simples e interpretável das árvores individuais é perdida no processo de agregação (ALTMAN; KRZYWINSKI, 2017).

Figura 8 – Princípio básico do *bagging*.

Fonte: (TIU et al., 2022)

3.4.4 Gradient Boosting Machines

O GBM (do inglês *Gradient Boosting Machine*) é uma técnica de aprendizado de máquina que combina múltiplos modelos fracos, geralmente árvores de decisão, para criar um modelo forte e preditivo. Ele funciona de forma sequencial, onde cada novo modelo é treinado para corrigir os erros do modelo anterior, minimizando iterativamente a função de perda (LANGO; BRZEZINSKI; STEFANOWSKI, 2016).

$$L(y, f(x)) \tag{10}$$

sendo:

y - O valor real ou esperado.

$f(x)$ - A saída ou previsão do modelo para a entrada x .

Em contraste com a técnica de *Roughly Balanced Bagging*, onde os modelos são construídos de forma independente, na técnica de *ensemble boosting* base do GBM, os modelos são criados sequencialmente, minimizando o erro dos modelos aprendidos anteriormente. O modelo preditivo é construído combinando os m modelos de árvores aditivos ($f_0, f_1, f_2, \dots, f_m$)

para prever os resultados conforme demonstrado na Equação 9. O modelo de conjunto de árvores é otimizado aplicando a redução do erro de generalização esperado L conforme a Equação 11 (ALZAMZAMI; HODA; SADDIK, 2020).

$$L = \sum_i^n (y_i - \hat{y}_i)^2 \quad (11)$$

sendo:

n - Número total de exemplos no conjunto de dados.

y_i - O valor real (esperado) para o exemplo i .

\hat{y}_i - A previsão do modelo para o exemplo i .

$(y_i - \hat{y}_i)^2$ - A diferença entre o valor real e a previsão, elevada ao quadrado.

O princípio central do GBM é criar modelos base que sejam altamente correlacionados com o gradiente negativo da função de perda associada ao conjunto completo. A função de perda, descrita na Equação 10, mede a discrepância entre os valores reais y e as previsões $f(x)$, e pode ser arbitrária, como o erro quadrático médio, mostrado na Equação 12.

$$L = \sum_{i=1}^n (y_i - f(x_i))^2 \quad (12)$$

sendo:

n - Número total de exemplos no conjunto de dados.

y_i - O valor real (esperado) para o exemplo i .

$f(x_i)$ - A previsão do modelo para o exemplo i .

O processo de aprendizado é iterativo e utiliza uma abordagem de descida de gradiente funcional, onde a atualização do modelo é dada pela Equação 13. Essa flexibilidade permite que os GBMs sejam altamente customizáveis, com a possibilidade de implementar funções de perda específicas para diferentes tarefas, como *quantile loss* para estimar distribuições condicionais. Além disso, os GBMs podem utilizar diferentes tipos de modelos base, como árvores de decisão, *splines* ou funções de base radial, permitindo capturar interações complexas entre variáveis (NATEKIN; KNOLL, 2013).

$$f_m(x) = f_{m-1}(x) + \rho_m g_m(x) \quad (13)$$

sendo:

$f_m(x)$ - Modelo atualizado na iteração.

$f_{m-1}(x)$ - Modelo da iteração anterior ($m - 1$).

ρ_m - Peso ou taxa de aprendizado que controla a contribuição do novo ajuste.

$g_m(x)$ - Função de ajuste na iteração m , que corrige os erros do modelo anterior.

O GBM, ao gerar modelos base sequencialmente, melhora a precisão das previsões ao focar em casos de treinamento difíceis de estimar. Cada modelo adicional tem como objetivo corrigir os erros dos modelos anteriores, com ênfase em amostras mal classificadas. O GBM utiliza uma abordagem de modelagem em estágios, minimizando uma função de perda, como erro quadrático ou absoluto, e é considerado uma forma de “gradiente funcional descendente”. A técnica é aprimorada com árvores de regressão de tamanho fixo, que particionam o espaço de entrada e predizem valores constantes para regiões específicas. O modelo de boosting pode alcançar um erro de treinamento muito baixo, mas o ajuste excessivo pode comprometer a capacidade de generalização. Parâmetros como a taxa de aprendizado e a complexidade da árvore influenciam o desempenho, sendo essencial encontrar um equilíbrio entre o número de iterações e a contribuição de cada modelo para evitar o sobreajuste. A escolha cuidadosa desses parâmetros é crucial para otimizar o desempenho do modelo (ZHANG; HAGHANI, 2015).

Segundo Zhang e Haghani (2015), o método de boosting surgiu como uma solução apresentada por Schapire (1990) para a questão levantada por Kearns (1988): é possível que um conjunto de aprendizes fracos seja equivalente a um único aprendiz forte? Um aprendiz fraco é definido como um algoritmo que tem um desempenho apenas um pouco superior ao de um palpite aleatório, enquanto um aprendiz forte é um modelo de previsão ou classificação significativamente mais preciso e bem alinhado ao problema. Essa questão é de grande relevância, pois, em geral, é mais simples desenvolver um modelo fraco do que um modelo forte.

O GBM estabelece os fundamentos teóricos que servem de base para diversos modelos, pois utiliza o gradiente descendente para ajustar modelos base de forma sequencial, oferece flexibilidade na escolha de funções de perda e permite personalizações para diferentes tarefas (NATEKIN; KNOLL, 2013). Modelos como o *Light Gradient Boosting Machine* (LightGBM) e o *Extreme Gradient Boosting* (XGBoost) são implementações específicas e otimizadas desse conceito de GBM.

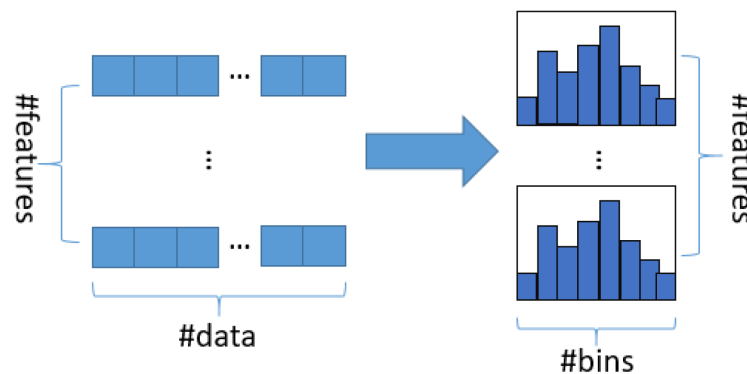
3.4.4.1 Light Gradient Boosting Machine (LightGBM)

O *Light Gradient Boosting Machine* (LightGBM), desenvolvido pela Microsoft em 2017 (KE et al., 2017), é um avançado algoritmo de *Gradient Boosting Decision Tree* (GBDT) que se destaca por seu método de crescimento de árvores baseado em folhas. Esse método prioriza as folhas que mais reduzem a perda, otimizando as divisões e tornando o treinamento

mais eficiente. O LightGBM foi projetado para lidar com os desafios de eficiência e escalabilidade em cenários com grandes volumes de dados e alta dimensionalidade de características. Utilizando um método baseado em histogramas, o algoritmo identifica os melhores candidatos a divisões, reduzindo a complexidade computacional sem comprometer a precisão.

A ideia básica do algoritmo de histogramas é discretizar características contínuas de ponto flutuante em k inteiros e construir um histograma com uma largura de k . Durante a passagem pelos dados, as estatísticas são acumuladas no histograma utilizando valor discretizado como índice. Após uma única passagem pelos dados, o histograma acumula o volume de estatísticas necessário, permitindo que o ponto de segmentação ótimo seja encontrado ao percorrer os valores discretos no histograma, como mostrado na Figura 9 (WANG; WANG, 2020).

Figura 9 – Diagrama esquemático do algoritmo do histograma.



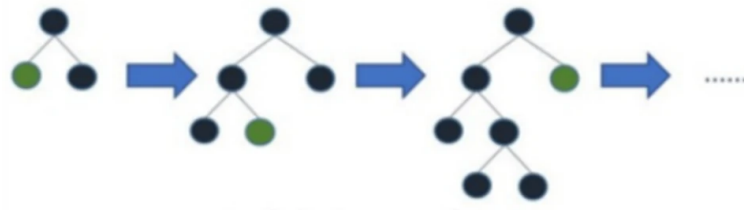
Fonte: (WANG; WANG, 2020).

Para aprimorar o treinamento, o LightGBM adota o algoritmo de amostragem *Gradient-based One-Side Sampling* (GOSS), que prioriza amostras com gradientes maiores, ignorando aquelas com gradientes pequenos, sob a premissa de que estas já apresentam erros reduzidos. No entanto, essa abordagem pode introduzir viés ao alterar a distribuição original dos dados. Para mitigar esse efeito, o GOSS realiza uma amostragem aleatória nos dados com gradientes pequenos, mantendo todas as amostras com gradientes grandes e ajustando seus pesos ao calcular o ganho de informação (ALZAMZAMI; HODA; SADDIK, 2020). Segundo Ke et al. (2017), o GOSS reduz o número de instâncias de dados ao priorizar aquelas com maiores gradientes, que têm maior impacto no cálculo do ganho de informação, enquanto descarta instâncias com gradientes menores. Isso permite uma estimativa precisa do ganho de informação com um conjunto de dados reduzido, otimizando o processo de treinamento.

Além disso, o LightGBM emprega o algoritmo *Exclusive Feature Bundling* (EFB) para lidar com a esparsidade dos dados, combinando características mutuamente exclusivas de

forma quase sem perdas. O EFB agrupa características que raramente possuem valores diferentes de zero simultaneamente em um único conjunto, reduzindo significativamente o número de características sem comprometer a precisão na determinação dos pontos de divisão (KE et al., 2017). Embora encontrar o agrupamento ideal de características seja um problema NP-difícil, o LightGBM utiliza um algoritmo guloso que em geral alcança uma boa aproximação. Essa combinação de técnicas, como o GOSS e o EFB, torna o LightGBM uma ferramenta poderosa e eficiente para análise de grandes volumes de dados, reduzindo o custo computacional e o consumo de memória em comparação com implementações tradicionais de GBDT, como o XGBoost (ALZAMZAMI; HODA; SADDIK, 2020).

Figura 10 – LightGBM pesquisa com crescimento da árvore por folha.



Fonte: (AZIZ et al., 2022).

O LightGBM é o algoritmo baseado em árvores mais rápido em comparação com outros, pois avança verticalmente. Sendo um algoritmo baseado em árvores, ele possui uma raiz e folhas que podem crescer vertical ou horizontalmente. A Figura 10 exemplifica o funcionamento do LightGBM: quando a pesquisa está na folha da esquerda (representada pelo círculo verde), em vez de ir para a folha mais à direita (círculo azul), o LightGBM se expande a partir da folha com maior perda significativa, crescendo verticalmente, ou seja, folha por folha. Em contraste, outros algoritmos crescem horizontalmente ou em níveis. Assim, o LightGBM é vantajoso ao processar grandes conjuntos de dados; no entanto, pode ocorrer *overfitting* em conjuntos menores. A principal vantagem desse algoritmo é que ele é de baixo custo computacional, consumindo uma quantidade extremamente baixa de memória para processar milhares de linhas, ao mesmo tempo que fornece resultados muito precisos (AZIZ et al., 2022).

Segundo Ke et al. (2017), uma característica proeminente do LightGBM é sua estrutura de *gradient boosting* com um classificador base sendo uma árvore de decisão. Para um conjunto de treinamento dado, conforme descrito na Equação 14, onde x é o conjunto de características e y é o rótulo, o algoritmo é inicializado com uma função constante $F_0 = \gamma$ e uma função de perda diferenciável.

$$X = \{(x_i, y_i) \mid x_i \in \mathbb{R}^k, y_i \in \mathbb{R}, k \geq 1, |X| = n\} \quad (14)$$

sendo:

X - O conjunto de pares (x_i, y_i) .

(x_i, y_i) - Cada elemento do conjunto é um par ordenado.

$x_i \in \mathbb{R}^2$ - x_i é um vetor em \mathbb{R}^2 (espaço bidimensional real).

$y_i \in \mathbb{R}$ - y_i é um valor real associado ao vetor x_i .

$k \geq 1$ - Condição adicional (não está diretamente relacionada aos pares, mas pode ser um parâmetro do problema).

$|X| = n$ - Tamanho do conjunto X é n , ou seja, ele contém n pares (x_i, y_i) .

Matematicamente, o LightGBM utiliza uma função de perda que combina o erro de predição e a regularização da complexidade das árvores, definida na Equação 15. O algoritmo utiliza a abordagem baseada em histogramas para encontrar os melhores pontos de divisão, agrupando valores contínuos em bins discretos, o que reduz a complexidade computacional. Além disso, o LightGBM implementa otimizações para lidar com dados esparsos, como a exclusão eficiente de valores ausentes no cálculo do ganho de perda. Essas melhorias permitem que o LightGBM seja até 20 vezes mais rápido que o GBDT convencional, mantendo uma precisão comparável (KE et al., 2017).

$$rL_{xgb} = \sum_{i=1}^N L(y_i, F(x_i)) + \sum_{m=1}^M \Omega(h_m) \quad (15)$$

onde:

$$\Omega(h) = \gamma T + \frac{1}{2} \lambda \|w\|^2$$

sendo:

T - número de folhas da árvore.

w - scores das folhas.

γ - controla a simplicidade das árvores.

λ - regula a magnitude dos pesos.

Os experimentos realizados com o LightGBM em diversos conjuntos de dados públicos, incluindo tarefas como classificação multiclasse, predição de cliques e aprendizado para ranqueamento, tem demonstrado sua superioridade em termos de velocidade de treinamento e consumo de memória. O algoritmo foi testado em cenários com características densas e esparsas, mostrando-se eficaz em ambos os casos. Além disso, o LightGBM utiliza estratégias de crescimento de árvores baseadas em folhas, o que melhora ainda mais sua eficiência.

3.4.4.2 Extreme Gradient Boosting (XGBoost)

Segundo Chen e Guestrin (2016), o XGBoost é amplamente utilizado por cientistas de dados para alcançar resultados de ponta em muitos desafios de aprendizado de máquina. O XGBoost é um método de *ensemble* baseado em árvores que implementa um framework de aprendizado de máquina por *gradient boosting* para problemas de regressão e classificação. O XGBoost cresce suas árvores utilizando algoritmos de divisão de forma nivelada e combina os resultados de maneira distinta. Inicialmente, algoritmos exatos e aproximados foram introduzidos para encontrar divisões, e posteriormente, um algoritmo baseado em histograma foi proposto após o desenvolvimento do algoritmo LightGBM. Este algoritmo constrói histogramas de características agrupando valores em “caixas”, permitindo que a divisão ocorra nas “caixas”, o que acelera o treinamento e reduz a complexidade computacional, salientando a eficiência dos métodos de *boosting* por gradiente. Adicionalmente, o XGBoost utiliza um algoritmo de divisão atento à qualidade do que se encontra disperso, eficaz em dados textuais esparsos, ignorando características nulas e alocando essas entradas conforme reduzir a perda (AZIZ et al., 2022).

O XGBoost destaca-se por lidar com dados esparsos usando um algoritmo baseado em esboço de quantil ponderado para aprendizado aproximado de árvores. Otimizado para padrões de acesso ao cache, compressão de dados e fragmentação, ele escala para bilhões de exemplos com menos recursos que sistemas tradicionais. Sua fórmula central minimiza uma função de perda regularizada, combinando erros quadráticos e regularização para evitar *overfitting* (CHEN; GUESTRIN, 2016). A escalabilidade é impulsionada por computação paralela e distribuída, com uso eficiente de memória. Desenvolvido como uma biblioteca de *boosting* de gradiente distribuído, o XGBoost é altamente versátil e portátil, com sua função objetivo descrita na Equação 16 (PUNURI et al., 2023).

$$F_t \approx \sum_{j=1}^T \left[\left(\sum_{i \in I_j} g_i \right) w_j + \frac{1}{2} \left(\sum_{i \in I_j} h_i + \lambda \right) w_j^2 \right] = \rho T \quad (16)$$

sendo:

F - função objetivo;

g_i - Primeira derivada do erro quadrático médio;

w - Pontuação de vetores nas folhas;

h_i - Segunda derivada do erro quadrático médio;

λ - Penalidade;

T - Número de folhas;

ρ - Complexidade da Folha;

I_j - Amostras de dados do nó folha j .

Segundo Zhang et al. (2021), o conceito fundamental do XGBoost consiste em incorporar continuamente árvores fracas, com diferentes pesos, ao conjunto. Essas árvores devem buscar se alinhar o mais próximo possível dos resíduos gerados pela previsão anterior. As Equações 17, 18 e ?? descrevem esse conceito. Na Equação 17, o valor previsto \hat{y}_i deve ser o mais próximo possível do valor verdadeiro, sem comprometer sua capacidade de generalização.

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i) \quad f_k \in F \quad (17)$$

sendo:

\hat{y} - é o valor previsto,

F - é o conjunto que inclui todas as árvores de regressão,

f_k - é uma das árvores de regressão e

K - é o número de árvores de regressão.

A fórmula para calcular o objetivo (Obj) está descrita na Equação 18, em que $l(y_i, \hat{y}_i^{(t)})$ refere-se à função de perda, que quantifica a discrepância entre o valor estimado e o valor real. Essa função pode assumir diversas formas, desde que seja derivável em segunda ordem. Por outro lado, Ω representa o termo de regularização, que determina a complexidade do modelo. Um valor menor de Ω indica uma complexidade reduzida, resultando em uma capacidade de generalização mais robusta.

$$\text{Obj}^{(t)} = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t)}) + \sum_{j=1}^t \Omega(f_j) + \text{constante} \quad (18)$$

Na Equação 19, T é o número de nós folha, e g é a pontuação representada pelos nós folha. O XGBoost utiliza a expansão de Taylor de segunda ordem para expandir a função de perda no processo de *boosting* por gradiente.

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \|w\|^2 \quad (19)$$

A Equação 20 descreve a função objetivo final, enquanto g_i e h_i são derivada de primeira ordem e a derivada de segunda ordem de cada ponto de dados na função de erro, e I_j é o conjunto de índices das amostras em cada nó folha.

$$\begin{aligned}
Obj^{(t)} &\simeq \sum_{i=1}^n \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \\
&= \sum_{i=1}^n \left[g_i w_q(x_i) + \frac{1}{2} h_i w_q^2(x_i) \right] + \gamma T + \lambda \frac{1}{2} \sum_{j=1}^T w_j^2 \\
&= \sum_{j=1}^T \left[\left(\sum_{i \in I_j} g_i \right) w_j + \frac{1}{2} \left(\sum_{i \in I_j} h_i + \lambda \sum_{j=1}^T \right) w_j^2 \right] + \gamma T \\
g_i &= \partial_{\hat{y}_{(t-1)}} l(y_i, \hat{y}_i^{(t-1)}), \quad h_i = \partial_{\hat{y}_{(t-1)}^2} l(y_i, \hat{y}_i^{(t-1)})
\end{aligned} \tag{20}$$

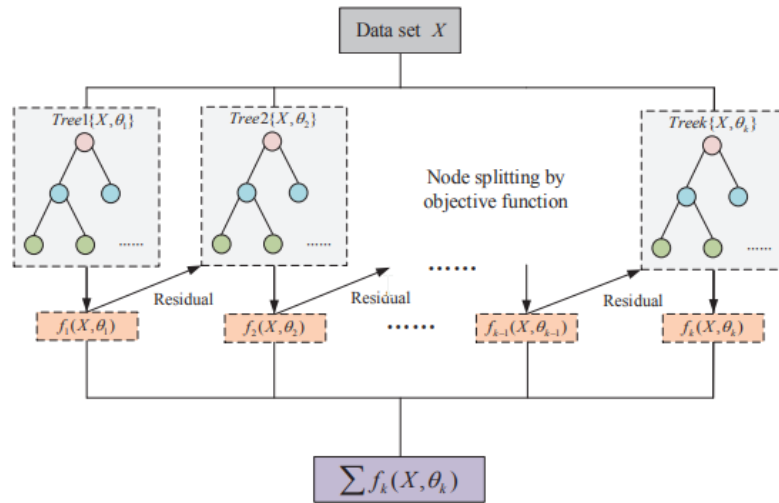
Outro parâmetro de regularização no XGBoost é o *shrinkage*, que reduz o tamanho do passo na expansão aditiva. Além disso, a complexidade das árvores pode ser limitada por outras estratégias, como a profundidade das árvores, entre outras. Uma vantagem secundária da redução da complexidade das árvores é que os modelos são treinados mais rapidamente e requerem menos espaço de armazenamento. Adicionalmente, o XGBoost implementa técnicas de randomização tanto para reduzir o overfitting quanto para aumentar a velocidade de treinamento. As técnicas de randomização incluídas no XGBoost são: amostras aleatórias para treinar árvores individuais e amostragem de colunas nos níveis de árvore e nó (BENTÉJAC; CSÖRGO; MARTÍNEZ-MUÑOZ, 2019).

A Figura 11, demonstra de forma esquemática o fluxograma do XGBoost, com o *dataset* de treinamento X (representado pelo retângulo cinza), durante o treinamento o modelo avalia continuamente a perda nos nós para identificar o nó folha com o maior ganho de perda (representados pelos círculos rosas, azuis e verde). O XGBoost constrói novas árvores dividindo repetidamente as características. Cada nova árvore (*Tree*) adicionada em uma iteração corresponde ao aprendizado de uma função $f_k(X, \Theta_k)$, que busca ajustar o resíduo da previsão anterior (retângulos de fundo azul). Após o treinamento, com K árvores geradas, as características das amostras de previsão estarão associadas a um nó folha em cada árvore (círculos verdes), e cada nó folha terá uma pontuação específica (retângulos laranjas). Por fim, as pontuações de todas as árvores são somadas para calcular o valor final de previsão da amostra (retângulo roxo) (GUO et al., 2020).

3.5 MÉTRICAS

Metodologias quantitativas e qualitativas foram empregadas neste trabalho para avaliar a eficácia prognóstica dos modelos. Dos experimentos realizados foram extraídas métricas como RMSE (do inglês *Root Mean Square Erro*), uma métrica estatística padrão amplamente utilizada para medir o desempenho de modelos. A Equação 21 define o RMSE, que penaliza mais os erros absolutos maiores e é adequado quando o erro segue uma distribuição normal. Essa métrica minimiza a soma dos erros quadráticos, representando bem a distribuição

Figura 11 – Diagrama esquemático do XGBoost.



Fonte: (GUO et al., 2020).

dos erros. Apesar de sua sensibilidade a *outliers*, o RMSE é eficaz para discriminar entre os resultados de diferentes modelos (CHAI; DRAXLER, 2014).

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (21)$$

sendo:

N - Total de amostras;

y_i - Valor real da observação;

\hat{y}_i - Valor calculado pelo modelo.

O *R-squared*, ou coeficiente de determinação R^2 , é uma métrica utilizada na análise de regressão para quantificar a proporção da variância da variável dependente explicada pelas variáveis independentes. O R^2 , cuja fórmula matemática é apresentada na Equação 22, é considerado mais informativo e confiável do que outras métricas, como SMAPE, MAPE, MAE, MSE e RMSE, especialmente em cenários onde a distribuição dos dados é relevante. O R^2 é útil em diversos domínios científicos para avaliar a qualidade de modelos de regressão, pois fornece uma medida clara de quão bem o modelo se ajusta aos dados observados. Apesar de suas limitações como a possibilidade de valores negativos, que não indicam a magnitude do mau desempenho, ele é recomendado como a métrica padrão para avaliação de regressões (CHICCO; WARRENS; JURMAN, 2021).

$$R^2 = \frac{\sum_{i=1}^N (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^N (y_i - \bar{y})^2} \quad (22)$$

sendo:

N - Total de amostras;

y_i - Valor real da observação;

\hat{y}_i - Valor calculado pelo modelo

\bar{y} - Média dos valores reais da observação y .

O *Mean Absolute Percentage Error* (MAPE) é uma métrica de avaliação de modelos de regressão que mede o erro percentual médio absoluto, sendo intuitiva para interpretar erros relativos. O MAPE, demonstrado na Equação 23, fornece uma compreensão clara dos erros em termos de porcentagens relativas, facilitando a interpretação do desempenho do modelo, especialmente em cenários onde as variações relativas são mais relevantes do que absolutas. O MAPE pode ser otimizado de forma semelhante ao MAE, utilizando uma estratégia de ponderação em regressão *kernel*. O MAPE é adequado para previsões em que a variável alvo é positiva e distante de zero, como em previsões de preços de bens de alto valor (MYTTENAERE et al., 2016).

$$MAPE = \frac{100}{N} \sum_{i=1}^N \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (23)$$

sendo:

N - Total de amostras;

y_i - Valor real da observação

\hat{y}_i - Valor calculado pelo modelo.

O *Mean Absolute Error* (MAE), ou erro médio absoluto, é outra métrica de avaliação de desempenho de modelos. O MAE, descrito na Equação 24, é considerado uma medida mais natural e direta do erro médio, pois expressa o erro em unidades da variável de interesse sem ser influenciado pela variabilidade dos erros ou pelo número de observações. Ele fornece uma representação clara do erro médio. O MAE deve ser utilizado em comparações de desempenho de modelos, principalmente quando não há um conjunto de estimativas considerado mais confiável (WILLMOTT; MATSUURA, 2005) .

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (24)$$

sendo:

n - Número total de observações;

y_i - Valor real da observação;

\hat{y}_i - Valor previsto pelo modelo.

Outra métrica relevante é o *Median Relative Absolute Error* (MdRAE), que mede o erro relativo absoluto mediano em relação a um modelo de referência. A fórmula do MdRAE é apresentada na Equação 25. Essa métrica é útil para avaliar o desempenho de modelos em comparação com um modelo de referência, especialmente em cenários onde a mediana é mais representativa do que a média, como em distribuições assimétricas.

$$MdRAE = \text{Mediana}_{i=1,n} \left(\frac{|y_i - \hat{y}_i|}{|y_i - \hat{b}_i|} \right) \quad (25)$$

sendo:

y_i - Valor - Valor real da observação, i .

\hat{y}_i - Previsão do modelo para a observação i .

\hat{b}_i - Previsão do modelo de referência (baseline) para a observação i .

$|y_i - \hat{y}_i|$ - Erro absoluto do modelo avaliado.

$|y_i - \hat{b}_i|$ - Erro absoluto do modelo de referência.

4 RESULTADOS E DISCUSSÕES

A análise dos resultados dos modelos de previsão, incluindo RF, XGBoost, LightGBM, KNN e SVM, mostrou que, mesmo sem ajustes de hiperparâmetros, esses modelos conseguem explicar uma grande parte da variância dos dados, como indicado pelos valores relativamente altos de R^2 (0,8864 para RF, 0,8104 para XGBoost, 0,8270 para LightGBM, e 0,7404 para KNN). Isso sugere que os modelos têm uma capacidade razoável de prever a temperatura do domo, destacando a eficácia dos algoritmos utilizados sob condições padrão em um contexto de validação cruzada K -fold de 5. No entanto, os valores elevados de RMSE para os modelos KNN e SVM (3,6329 e 4,7619, respectivamente) indicam que, apesar da boa explicação da variância, as previsões podem ser imprecisas em termos absolutos, sugerindo a ocorrência de erros significativos nas previsões. Essa congruência entre R^2 e RMSE alto indica a necessidade de ajustes nos hiperparâmetros dos modelos para melhorar a precisão das previsões.

As Tabelas 4 apresenta o resultado das métricas R^2 e RMSE dos métodos investigados, antes e depois do ajuste dos hiperparâmetros, respectivamente. Esses ajustes foram obtidos utilizando a validação cruzada k -fold e *GridSearch* com busca por gradiente. Em linhas gerais, as tabelas mostram uma melhoria no RMSE para a maioria dos modelos. Apesar de ser uma melhoria pequena, ela é significativa no contexto da fabricação do ferro gusa, onde otimizações aparentemente mínimas podem impactar significativamente a eficiência e a qualidade do processo produtivo dado o volume da produção, ajustes e a quantificação do ganho demonstram a eficácia das técnicas de otimização de hiperparâmetros. Por fim, como os métodos XGB e LB apresentaram os melhores resultados, a análise desses métodos será aprofundada a seguir.

Tabela 4 – Performance dos Modelos, antes do ajuste dos hiperparâmetros.

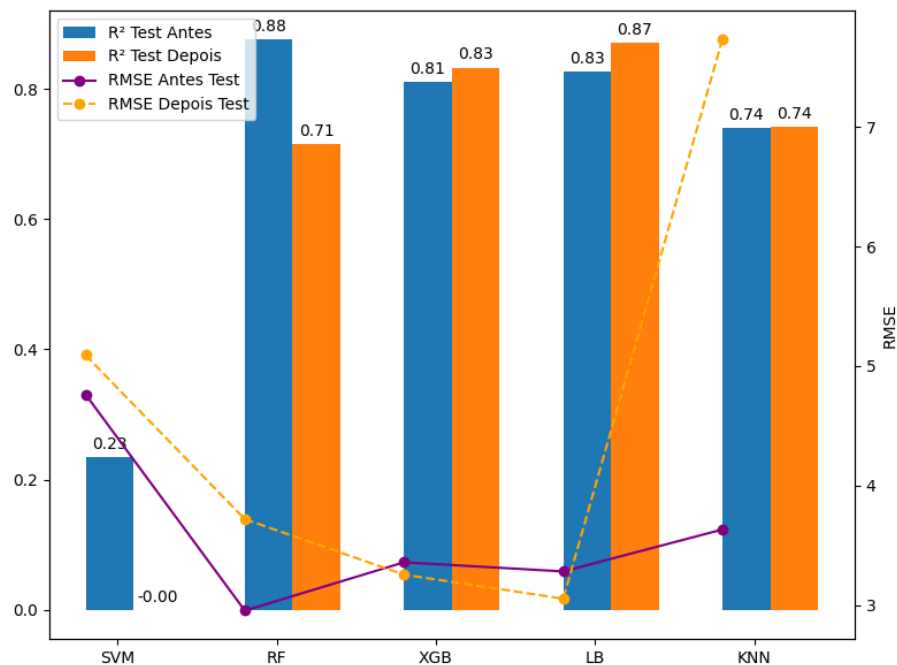
| Modelos | Antes do hiperparâmetros | | | | Depois do hiperparâmetros | | | |
|---------|--------------------------|--------|-----------|--------|---------------------------|---------|-----------|--------|
| | R^2 | | RMSE (°C) | | R^2 | | RMSE (°C) | |
| | Train | Test | Train | Test | Train | Test | Train | Test |
| SVM | 0,2628 | 0,2338 | 4,6180 | 4,7619 | 0,9508 | -0,0012 | 2,3467 | 5,0914 |
| RF | 0,9771 | 0,8864 | 1,9376 | 2,9544 | 0,8564 | 0,7148 | 3,0679 | 3,7194 |
| XGB | 0,9973 | 0,8104 | 1,1309 | 3,3585 | 0,9697 | 0,8327 | 2,0785 | 3,2550 |
| LB | 0,9442 | 0,8270 | 2,4215 | 3,2821 | 0,9856 | 0,8704 | 1,7254 | 3,0538 |
| KNN | 0,9086 | 0,7404 | 2,7399 | 3,6329 | 0,8845 | 0,7414 | 9,8953 | 7,7346 |

Fonte: Próprio autor.

A Figura 12 ilustra graficamente os resultados do R^2 e do RMSE dos métodos investigados. As barras azuis representam o resultado da métrica R^2 antes do ajuste dos hiperparâmetros, enquanto as barras laranja mostram o resultado após os ajustes. A linha contínua roxa indica o RMSE dos testes de cada método antes do ajuste dos hiperparâmetros, enquanto a linha pontilhada amarela identifica o RMSE depois dos ajustes. No gráfico, é

possível observar nas duas primeiras barras o método SVM, destacando que o ajuste de hiperparâmetros não melhorou seu desempenho, resultando em um aumento no RMSE e uma redução no R^2 . As terceira e quarta barras referem-se ao RF, cujo desempenho piorou após o ajuste, com um aumento no RMSE e uma redução no R^2 de 0,89 para 0,71. As duas próximas barras referem-se ao XGB, onde se observa uma ligeira melhoria no desempenho, com uma pequena redução no RMSE e um pequeno aumento no R^2 de 0,81 para 0,83. As duas barras seguintes correspondem ao LightGBM, para o qual o ajuste de hiperparâmetros melhorou o desempenho, reduzindo o RMSE e aumentando o R^2 de 0,83 para 0,87. Finalmente, as duas últimas barras dizem respeito ao KNN, que apresentou uma piora significativa no desempenho após o ajuste, com um grande aumento no RMSE e estabilidade no R^2 de 0,74.

Figura 12 – Comparação R^2 e RMSE dos testes antes e depois dos hiperparâmetros.

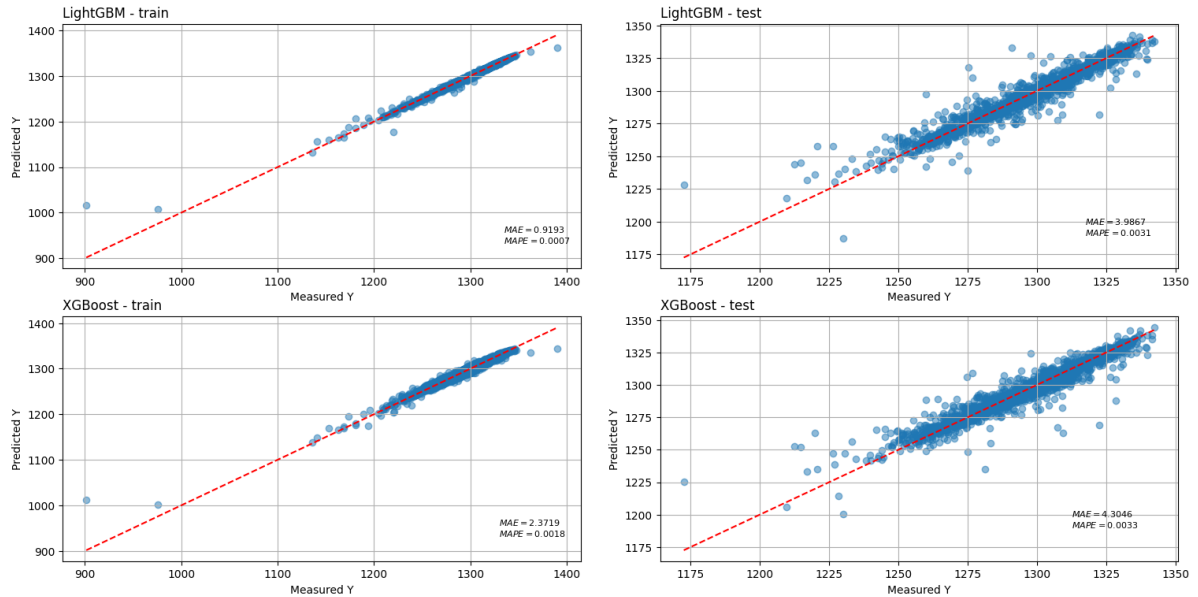


Fonte: Próprio autor.

A Figura 13 apresenta os gráficos de dispersão que comparam as temperaturas de domo previstas pelos modelos XGBoost e LightGBM com as temperaturas medidas, tanto na fase de treinamento quanto de teste. Nos gráficos, a linha vermelha pontilhada indica erro 0 ou seja uma predição perfeita da temperatura de domo, enquanto os pontos azuis indicam a predição obtida pelos métodos analisados. Adicionalmente, à esquerda estão os experimentos de treinamento e à direita estão os experimentos de teste, enquanto os gráficos superiores são gerados pelo LightGBM e os inferiores pelo XGBoost. Os resultados indicam uma forte correlação dos modelos na predição das temperaturas de domo com a temperatura ideal a ser atingida. Nos gráficos de teste (à direita), embora as temperaturas de domo previstas estejam mais dispersas, em sua maioria ainda se encontram próximas à

linha de predição ideal, sugerindo que os modelos foram capazes de prever as tendências de temperaturas esperadas no domo.

Figura 13 – Gráficos de dispersão comparando a predição dos métodos XGBoost e LightGBM.



Fonte: Próprio autor.

A Figura 14 apresenta uma comparação do desempenho dos modelos LightGBM (acima) e XGBoost (abaixo), destacando as predições dos modelos (linha pontilhada vermelha) em relação aos dados reais de temperatura (linha azul). Ao observar a linha azul, nota-se que os dados apresentam uma distribuição aparentemente aleatória, sem padrões consistentes ou tendências claras. Dessa forma, os dados não podem ser caracterizados como uma série temporal, pois não exibem dependência temporal ou comportamento previsível ao longo do tempo, o que justifica a investigação dos métodos analisados nesse trabalho. Adicionalmente, como na Figura 13, a esquerda estão os resultados do treinamento e a direita os resultados do teste. Como esperado, nos gráficos de treinamento de ambos os modelos, foi observada uma alta precisão na fase de treino, evidenciada pela sobreposição da linha pontilhada vermelha com a linha contínua azul.

Tabela 5 – Resultado das métricas MAE e MAPE para os modelos LightGBM e XGBoost.

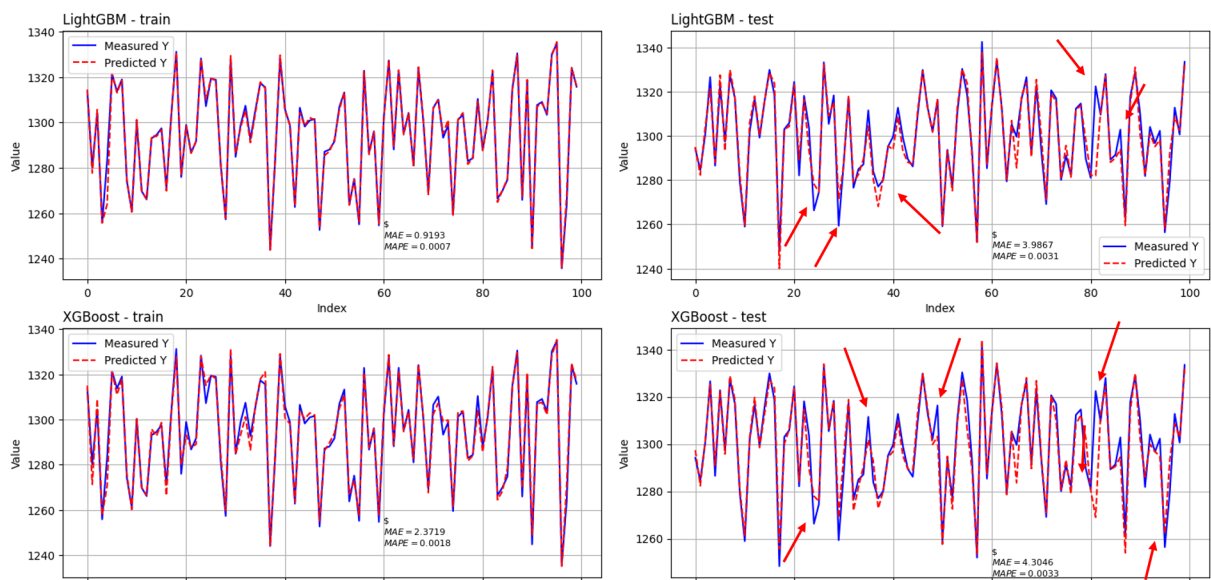
| Modelos | MAE | | MAPE | |
|----------|--------|--------|--------|--------|
| | Train | Test | Train | Test |
| LightGBM | 0,9193 | 3,9867 | 0,0007 | 0,0031 |
| XGBoost | 2,3719 | 4,3046 | 0,0018 | 0,0033 |

Fonte: Próprio autor.

Nos gráficos de teste da Figura 14, observa-se que em algumas comparações as linhas não se sobrepõem perfeitamente, indicando erros pontuais na predição indicados pelas setas vermelhas. Essas discrepâncias são notadas apenas em situações onde os dados

sugerem uma inversão na tendência da temperatura do domo, possivelmente devido a mudanças entre as fases de *on-gas* e *on-blast*. Este padrão nos gráficos de linha sinaliza áreas específicas onde os modelos podem necessitar de ajustes ou melhorias para lidar com as dinâmicas de mudança de fase, visando aprimorar a precisão das previsões. Ainda sobre os gráficos de teste, visualmente, é quase imperceptível que no gráfico LightGBM-Test há menos desvios em relação à linha azul do que no gráfico XGBoost-Test. Assim, para uma comparação mais precisa apresentamos a Tabela 5 com as métricas MAE e MAPE dessas comparações. Pela tabela observamos que o modelo LightGBM tem um desempenho ligeiramente melhor do que o XGBoost.

Figura 14 – Comparação da predição dos métodos XGBoost e LightGBM



Fonte: Próprio autor.

5 CONCLUSÕES E TRABALHOS FUTUROS

Esse trabalho abrangeu um estudo completo para analisar o desempenho de métodos de aprendizado de máquina na previsão da temperatura de regeneradores em uma usina siderúrgica integrada. Os métodos de previsão selecionados para essa investigação foram escolhidos levando em consideração que os dados utilizados não configuram uma série temporal. Dessa forma, os métodos investigados foram os *Support Vector Machine* (SVM), *Random Forest* (RF), *Extreme Gradient Boosting* (XGBoost), *Light Gradient Boosting Machine* (LightGBM) e *K-Nearest Neighbors*(KNN).

Seguindo o roteiro de uma análise de previsão completa, inicialmente foi feita uma seleção de atributos. Para isso, foram realizadas entrevistas com especialistas, seguidas pela aplicação do método de Análise de Componentes Principais (PCA). Das 94 variáveis originais, 16 foram selecionadas para prosseguir com a investigação. Posteriormente, os *outliers* foram detectados e tratados utilizando a técnica de *IQR*, os dados faltantes identificados foi usada a média móvel das últimas 12 amostras, em seguida a técnica *Min-Max Scaling* foi utilizada para normalização dos dados em um intervalo de 0 e 1, por fim procedeu-se à filtragem dos dados.

Com o objetivo de aprimorar o desempenho das previsões, foi empregada a ferramenta de ajuste de parâmetros dos métodos, denominada *GridSearchCV*. Através da avaliação das métricas R^2 e RMSE, foi destacado o impacto da calibração dos parâmetros na qualidade das previsões. O ajuste dos parâmetros apresentou melhora na qualidade das previsões e das métricas R^2 e RMSE, conforme demonstrado nas tabelas 4, resultado no melhor desempenho dos métodos, destacando-se o XGBoost e LightGBM como melhor desempenho.

Explorando os modelos que obtiveram os melhores desempenhos nas métricas R^2 e RMSE, foi feita uma análise visual dos gráficos de comparação, incluindo o *scatter plots* da temperatura ideal teórica com as previsões e uma comparação da temperatura medida com a prevista. Essa análise revelou que ambos os modelos, XGBoost e LightGBM, proporcionam aplicação precisas, embora o LightGBM aparentemente apresente uma ligeira vantagem. Para quantificar essa vantagem foi apresentado as métricas MAE e MAPE desses métodos.

Trabalhos futuros poderia incluir esses resultados em métodos de otimização visando, entre outros objetivos, reduzir o consumo de gases utilizados na queima para aquecimento do regenerador, ampliando assim a eficiência operacional na usina siderúrgica, também a elaboração de simuladores operacionais para cenários buscando ampliar o ciclo *on-blast*.

REFERÊNCIAS

- ALIFIA, Tiara Ratu et al. Model algoritma knn untuk prediksi kelulusan mahasiswa stikom cki. *Jurnal Ilmiah Informatika Komputer*, Gunadarma University, v. 29, n. 2, p. 167–181, 2023.
- ALSHBOUL, Odey et al. A comparative study of lightgbm, xgboost, and gep models in shear strength management of sfrc-sbws. In: ELSEVIER. *Structures*. [S.l.], 2024. v. 61, p. 106009.
- ALTMAN, Naomi; KRZYWINSKI, Martin. Ensemble methods: bagging and random forests. *Nature Methods*, Nature Publishing Group, v. 14, n. 10, p. 933–935, 2017.
- ALZAMZAMI, Fatimah; HODA, Mohamad; SADDIK, Abdulmotaleb El. Light gradient boosting machine for general sentiment classification on short texts: a comparative evaluation. *IEEE access*, IEEE, v. 8, p. 101840–101858, 2020.
- AZIZ, Rabia Musheer et al. Lgbm: a machine learning approach for ethereum fraud detection. *International Journal of Information Technology*, Springer, v. 14, n. 7, p. 3321–3331, 2022.
- BENTÉJAC, Candice; CSÖRGO, Anna; MARTÍNEZ-MUÑOZ, Gonzalo. A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, v. 54, p. 1937 – 1967, 2019. Disponível em: <<https://api.semanticscholar.org/CorpusID:221283893>>.
- BING-YAO, Cai; HUI, Zhao; YOU-JUN, Yue. Research on the bof steelmaking endpoint temperature prediction. In: IEEE. *2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC)*. [S.l.], 2011. p. 2278–2281.
- BRASIL, Instituto Aço. *Mercado Brasileiro do Aço - Análise setorial e regional*. 2021. 6-7 p. Disponível em: <https://acobrasil.org.br/site/wp-content/uploads/2021/08/MBA_Ediãgãčo-2021.pdf>. Acesso em: 28 novembro 2021.
- BRASIL, Instituto Aço. *Mercado Brasileiro do Aço - Análise setorial e regional*. 2023. 6-7 p. Disponível em: <https://acobrasil.org.br/site/wp-content/uploads/2023/11/Estatistica-da-Siderurgia_3o_TRI_2023.pdf>. Acesso em: 28 novembro 2021.
- BRASIL, Portal Siderurgia. *Anuário Brasileiro da Siderurgia 2021*. 2020. 12 p. Disponível em: <https://siderurgiabrasil.com.br/wp-content/uploads/2020/05/gc2021_site.pdf>. Acesso em: 22 julho 2021.
- BREIMAN, Leo. Bagging predictors. *Machine learning*, Springer, v. 24, n. 2, p. 123–140, 1996.
- CARMO, Marcelo; MUTZ, Filipe; RESENDO, Leandro. Improving steel making off-gas predictions by mixing classification and regression multi-modal multivariate models. In: *Anais do XIX Encontro Nacional de Inteligência Artificial e Computacional*. Porto Alegre, RS, Brasil: SBC, 2022. p. 37–48. ISSN 2763-9061.
- CHAI, Tianfeng; DRAXLER, Roland R. Root mean square error (rmse) or mean absolute error (mae)?—arguments against avoiding rmse in the literature. *Geoscientific model development*, Copernicus Publications Göttingen, Germany, v. 7, n. 3, p. 1247–1250, 2014.

- CHEN, Tianqi; GUESTRIN, Carlos. Xgboost: A scalable tree boosting system. In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. [S.l.: s.n.], 2016. p. 785–794.
- CHICCO, Davide; WARRENS, Matthijs J; JURMAN, Giuseppe. The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation. *Peerj computer science*, PeerJ Inc., v. 7, p. e623, 2021.
- CHOI, Hyun-Jun et al. Predicting frost depth of soils in south korea using machine learning techniques. *Sustainability*, MDPI, v. 14, n. 15, p. 9767, 2022.
- DANG, Yijie et al. Image classification based on quantum k-nearest-neighbor algorithm. *Quantum Information Processing*, Springer, v. 17, p. 1–18, 2018.
- DENG, Xinyang et al. An improved method to construct basic probability assignment based on the confusion matrix for classification problem. *Information Sciences*, Elsevier, v. 340, p. 250–261, 2016.
- DIETTERICH, Tom. Overfitting and undercomputing in machine learning. *ACM computing surveys (CSUR)*, ACM New York, NY, USA, v. 27, n. 3, p. 326–327, 1995.
- DING, Shenyi et al. Dynamic prediction of the silicon content in the blast furnace using lstm-rnn-based models. In: IEEE. *2018 International Computers, Signals and Systems Conference (ICOMSSC)*. [S.l.], 2018. p. 491–495.
- DONG, Xibin et al. A survey on ensemble learning. *Frontiers of Computer Science*, Springer, v. 14, n. 2, p. 241–258, 2020.
- FARAHNAKIAN, Fahimeh et al. Energy aware consolidation algorithm based on k-nearest neighbor regression for cloud data centers. In: IEEE. *2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing*. [S.l.], 2013. p. 256–259.
- GENG, Shu. Analysis of the different statistical metrics in machine learning. *Highlights in Science Engineering and Technology*, Darcy Roy Press Co. Ltd., v. 88, p. 350–356, 2024.
- GHOSH, Anil K. On optimum choice of k in nearest neighbor classification. *Computational Statistics & Data Analysis*, Elsevier, v. 50, n. 11, p. 3113–3123, 2006.
- GHOSH, Sourish; DASGUPTA, Anasuya; SWETAPADMA, Aleena. A study on support vector machine based linear and non-linear pattern classification. In: IEEE. *2019 International Conference on Intelligent Sustainable Systems (ICISS)*. [S.l.], 2019. p. 24–28.
- GUO, Gongde et al. Knn model-based approach in classification. In: SPRINGER. *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings*. [S.l.], 2003. p. 986–996.
- GUO, Rui et al. Degradation state recognition of piston pump based on iceemdan and xgboost. *Applied Sciences*, MDPI, v. 10, n. 18, p. 6593, 2020.
- HASTIE, Trevor et al. *The elements of statistical learning: data mining, inference, and prediction*. [S.l.]: Springer, 2009. v. 2.
- HEATH, D; KASIF, S; SALZBERG, S. k-dt: A multi-tree learning method. In: *Proc. of the Second Int. Workshop on Multistrategy Learning*. [S.l.: s.n.], 1993. p. 138–149.

- JAMES, Gareth et al. *An introduction to statistical learning*. [S.l.]: Springer, 2013. v. 112.
- KACUR, Jan et al. Prediction of temperature and carbon concentration in oxygen steelmaking by machine learning: A comparative study. *Applied Sciences*, v. 12, n. 15, p. 7757–7757, 2022.
- KE, Guolin et al. Lightgbm: A highly efficient gradient boosting decision tree. In: *Neural Information Processing Systems*. [s.n.], 2017. Disponível em: <<https://api.semanticscholar.org/CorpusID:3815895>>.
- KEARNS, Michael. Thoughts on hypothesis boosting. *Unpublished manuscript*, v. 45, p. 105, 1988.
- KOYFMAN, Oleksiy et al. Using of intelligence analysis of technological parameters database for implementation of control subsystem of hot blast stoves block acs. In: *CMIS*. [S.l.: s.n.], 2021. p. 145–157.
- KRAMER, Oliver et al. *Dimensionality reduction with unsupervised nearest neighbors*. [S.l.]: Springer, 2013. v. 51.
- KWAKERNAAK, H; TIJSEN, P; STRIJBOS, RCW. Optimal operation of blast furnace stoves. *Automatica*, Elsevier, v. 6, n. 1, p. 33–40, 1970.
- LANGO, Mateusz; BRZEZINSKI, Dariusz; STEFANOWSKI, Jerzy. Put at semeval-2016 task 4: The abc of twitter sentiment analysis. In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. [S.l.: s.n.], 2016. p. 126–132.
- LEE, Tae-Hwy; ULLAH, Aman; WANG, Ran. Bootstrap aggregating and random forest. In: *Macroeconomic forecasting in the era of big data*. [S.l.]: Springer, 2020. p. 389–429.
- MATINO, Ismael et al. Two innovative modelling approaches in order to forecast consumption of blast furnace gas by hot blast stoves. *Energy Procedia*, Elsevier, v. 158, p. 4043–4048, 2019.
- MYTTENAERE, Arnaud De et al. Mean absolute percentage error for regression models. *Neurocomputing*, Elsevier, v. 192, p. 38–48, 2016.
- NATEKIN, Alexey; KNOLL, Alois. Gradient boosting machines, a tutorial. *Frontiers in neurorobotics*, Frontiers Media SA, v. 7, p. 21, 2013.
- NAVARRO, Leticia C et al. Temperature prediction in blast furnaces: a machine learning comparative study. In: IEEE. *2022 35th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. [S.l.], 2022. v. 1, p. 192–197.
- OBERGASSEL, Wolfgang et al. Phoenix from the ashes: an analysis of the paris agreement to the united nations framework convention on climate change; part 1. 2015.
- ORTIZ-IMEDIO, Rafael; ORTIZ, Alfredo; ORTIZ, Inmaculada. Comprehensive analysis of the combustion of low carbon fuels (hydrogen, methane and coke oven gas) in a spark ignition engine through cfd modeling. *Energy Conversion and Management*, Elsevier, v. 251, p. 114918, 2022.
- PUNURI, Sudheer Babu et al. Efficient net-xgboost: an implementation for facial emotion recognition using transfer learning. *Mathematics*, MDPI, v. 11, n. 3, p. 776, 2023.

- RIZZO, Ernandes Marcos da Silveira. *Processo de fabricação de ferro-gusa em alto-forno*. [S.l.]: ABM, 2009.
- ŞAHİN, Akın; MORARI, Manfred. Optimizing control of hot blast stoves in staggered parallel operation. *IFAC Proceedings Volumes*, Elsevier, v. 41, n. 2, p. 1982–1987, 2008.
- SALAZAR, Jose J et al. Fair train-test split in machine learning: Mitigating spatial autocorrelation for improved prediction accuracy. *Journal of Petroleum Science and Engineering*, Elsevier, v. 209, p. 109885, 2022.
- SALMAN, Hasan Ahmed; KALAKECH, Ali; STEITI, Amani. Random forest algorithm overview. *Babylonian Journal of Machine Learning*, v. 2024, p. 69–79, 2024.
- SCHAPIRE, Robert E. The strength of weak learnability. *Machine learning*, Springer, v. 5, p. 197–227, 1990.
- SHATOKHA, Volodymyr. Environmental sustainability of the iron and steel industry: Towards reaching the climate goals. *European Journal of Sustainable Development*, v. 5, n. 4, p. 289–289, 2016.
- SUBRAMANIASWAMY, V; LOGESH, R. Adaptive knn based recommender system through mining of user preferences. *Wireless Personal Communications*, Springer, v. 97, p. 2229–2247, 2017.
- SUI, Peng et al. Current situation and development prospects of metallurgical by-product gas utilization in china's steel industry. *International Journal of Hydrogen Energy*, Elsevier, v. 48, n. 74, p. 28945–28969, 2023.
- SUN, Wenqiang; WANG, Zihao; WANG, Qiang. Hybrid event-, mechanism-and data-driven prediction of blast furnace gas generation. *Energy*, Elsevier, v. 199, p. 117497, 2020.
- TANG, Yao Wu; LIU, Xiang. Hot blast stove temperature control system based on neural network predictive control. In: *Designing and Researching of Machines and Technologies for Modern Manufacture*. [S.l.]: Trans Tech Publications Ltd, 2015. (Applied Mechanics and Materials, v. 709), p. 281–284.
- TEHRANI, Sara Saberi Moghadam et al. Visual transformer and deep cnn prediction of high-risk covid-19 infected patients using fusion of ct images and clinical data. *BMC Medical Informatics and Decision Making*, Springer, v. 23, n. 1, p. 265, 2023.
- TIU, Ervin et al. An evaluation of various data pre-processing techniques with machine learning models for water level prediction. *Natural Hazards*, v. 110, 01 2022.
- WANG, Yan; WANG, Tao. Application of improved lightgbm model in blood glucose prediction. *Applied Sciences*, MDPI, v. 10, n. 9, p. 3227, 2020.
- WILLMOTT, Cort J.; MATSUURA, Kenji. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate Research*, v. 30, p. 79–82, 2005. Disponível em: <<https://api.semanticscholar.org/CorpusID:120556606>>.
- WU, Xiao et al. Power-carbon coordinated control of bfg-fired ccgt power plant integrated with solvent-based post-combustion co2 capture. *Energy*, Elsevier, v. 226, p. 120435, 2021.

- XIE, Shengkun; KRISHNAN, Sridhar et al. Dynamic principal component analysis with nonoverlapping moving window and its applications to epileptic eeg classification. *The Scientific World Journal*, Hindawi, v. 2014, 2014.
- XING, Wenchao; BEI, Yilin. Medical health big data classification based on knn classification algorithm. *Ieee Access*, IEEE, v. 8, p. 28808–28819, 2019.
- XUE, Liang et al. A data-driven shale gas production forecasting method based on the multi-objective random forest regression. *Journal of Petroleum Science and Engineering*, Elsevier, v. 196, p. 107801, 2021.
- YANG, Yinghua; YAN, Zhiwen; LIU, Xiaozhi. A gru-based time-space reconstruction network for dome temperature prediction of hot blast stove. In: IEEE. *2024 36th Chinese Control and Decision Conference (CCDC)*. [S.l.], 2024. p. 1471–1476.
- YANG, Yinghua; ZHAO, Xin; LIU, Xiaozhi. A novel echo state network and its application in temperature prediction of exhaust gas from hot blast stove. *IEEE Transactions on Instrumentation and Measurement*, IEEE, v. 69, n. 12, p. 9465–9476, 2020.
- ZHANG, Haotian; ZHANG, Lin; JIANG, Yuan. Overfitting and underfitting analysis for deep learning based end-to-end communication systems. In: IEEE. *2019 11th international conference on wireless communications and signal processing (WCSP)*. [S.l.], 2019. p. 1–6.
- ZHANG, Lingyu et al. Time series forecast of sales volume based on xgboost. In: IOP PUBLISHING. *Journal of Physics: Conference Series*. [S.l.], 2021. v. 1873, n. 1, p. 012067.
- ZHANG, Shichao. Nearest neighbor selection for iteratively knn imputation. *Journal of Systems and Software*, Elsevier, v. 85, n. 11, p. 2541–2552, 2012.
- ZHANG, Shichao. Challenges in knn classification. *IEEE Transactions on Knowledge and Data Engineering*, IEEE, v. 34, n. 10, p. 4663–4675, 2021.
- ZHANG, Shichao et al. Efficient knn classification with different numbers of nearest neighbors. *IEEE transactions on neural networks and learning systems*, IEEE, v. 29, n. 5, p. 1774–1785, 2017.
- ZHANG, Yanru; HAGHANI, Ali. A gradient boosting method to improve travel time prediction. *Transportation Research Part C: Emerging Technologies*, Elsevier, v. 58, p. 308–324, 2015.
- ZHOU, Heng; ZHANG, Haifeng; YANG, Chunjie. Hybrid-model-based intelligent optimization of ironmaking process. *IEEE Transactions on Industrial Electronics*, IEEE, v. 67, n. 3, p. 2469–2479, 2019.
- ZHU, Yuqing et al. Private-knn: Practical differential privacy for computer vision. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2020. p. 11854–11862.