UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Sensor virtual analisador de oxigênio residual para forno de cal em indústria de celulose baseado em Inteligência Artificial

Khayan Sobral Marques

Trabalho de Conclusão de Curso
MBA em Inteligência Artificial e Big Data

ICMC USP SÃO CARLOS

UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Sensor virtual analisador de oxigênio residual para forno de cal em indústria de celulose baseado em Inteligência Artificial

Khayan Sobral Marques

USP - São Carlos

2022

Khayan Sobral Marques

Sensor virtual analisador de oxigênio residual para forno de cal em indústria de celulose baseado em Inteligência Artificial

Trabalho de conclusão de curso apresentado ao Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, como parte dos requisitos para obtenção do título de Especialista em Inteligência Artificial e Big Data.

Área de concentração: Inteligência Artificial

Orientador: Prof. Dr. Ricardo Rodrigues Ciferri

USP - São Carlos

2022

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

M357s

Marques, Khayan Sensor virtual analisador de oxigênio residual para forno de cal em indústria de celulose baseado em Inteligência Artificial / Khayan Marques; orientador Ricardo Ciferri. -- São Carlos, 2022. 58 p.

Trabalho de conclusão de curso (MBA em Inteligência Artificial e Big Data) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2022.

1. sensor virtual. 2. forno de cal. 3. oxigênio residual. 4. inteligência artificial. 5. aprendizado de máquina. I. Ciferri, Ricardo, orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

DEDICATÓRIA

Dedico este trabalho a Deus e toda minha família Ana Neide, Manoel e Thaynan que sempre estiveram presente em minha vida, me apoiando e suportando em todos os momentos. Dedico também a minha noiva Carolina Berdague que sempre esteve comigo em todos os momentos, suportando e me ajudando a seguir em frente. Deixo minha dedicatória também para o meu amigo e colega de trabalho Kleverson Glauber, sem você nada disso seria possível, Obrigado.

AGRADECIMENTOS

Ao Sr. Kleverson Glauber, pelo incentivo, apoio, aulas, discussões e todos os momentos em que me fez repensar e crescer como pessoa e profissional.

Aos Senhores Luis Binotto e Henrique Garcia por apoiar e incentivar fortemente esta etapa profissional.

Ao Dr. Ricardo Ciferri, por todo o apoio durante o curso.

EPÍGRAFE

“Ever try.
Ever fail.
No matter.
Try again.
Fail again.
Fail better”
Samuel Beckett (1983)

RESUMO

MARQUES, K. S. **Sensor virtual analisador de oxigênio residual para forno de cal em indústria de celulose baseado em Inteligência Artificial.** 2022. 52 f. Trabalho de conclusão de curso (MBA em Inteligência Artificial e Big Data) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2022.

A alta e crescente demanda global no consumo de celulose, tem desafiado às indústrias do setor a serem mais eficientes na utilização dos seus recursos, aplicando cada vez mais otimização nos seus processos. A etapa de recuperação de cal é fundamental para redução vertiginosa do custo de produção, uma vez que essa cal é reutilizada para formação do principal insumo químico utilizado na planta. O equipamento responsável para a recuperação dessa lama de cal oriunda do processo é o forno de cal. Se tratando de um processo de combustão com múltiplos combustíveis, onde dois deles são subprodutos do processo e o terceiro sendo gás natural de petróleo adquirido externamente. É necessário reduzir o consumo desse combustível externo, o terceiro maior custo variável da planta, sendo assim fundamental a otimização da combustão ocorrida nessa área, sem comprometer a qualidade da cal recuperada. Atualmente é utilizado um robô para realizar a medição contínua e em tempo real do oxigênio residual dentro do forno, a qual é utilizada como variável do processo em um controle regulatório clássico *feedback* que manipula um ventilador de exaustão de gases a fim de manter a variável do processo no valor desejado. A proposta deste trabalho é virtualizar este robô analisador de oxigênio residual visando a redução da variabilidade do oxigênio, através do uso de inteligência artificial, aplicando diferentes tipos de algoritmos para construção do modelo preditivo, buscando aquele de melhor acurácia e poder de generalização baseado nos indicadores de performance *RMSE* para acurácia e coeficiente de determinação para poder de generalização. Foram comparados algoritmos de aprendizado de máquina árvore de decisão, florestas aleatórias, *XGBoost* e *Lightgbm*.

Palavras-chave: celulose; forno de cal; sensor virtual; oxigênio residual; inteligência artificial; aprendizado de máquina.

ABSTRACT

MARQUES, K. S. **Virtual sensor residual oxygen analyzer for lime kiln in cellulose industry based on Artificial Intelligence.** 2022. 52 f. Trabalho de conclusão de curso (MBA em Inteligência Artificial e Big Data) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2022.

The high and growing global demand for pulp consumption has challenged industries in the sector to be more efficient in the usage of their resources, applying more and more optimization in their processes. The lime recovery step is essential for a dramatic reduction in production costs, since this lime is reused to form the main chemical input used in the plant. The equipment responsible for recovering this lime mud from the process is the lime kiln. This is a combustion process with multiple fuels, where two of them are by-products of the process and the third one is a petroleum-based natural gas acquired externally. It is necessary to reduce the consumption of this external fuel, the third largest variable cost of the plant, thus it is essential to optimize the combustion that takes place in this area, without compromising the quality of the recovered lime. A robot is currently used to perform continuous real-time measurement of residual oxygen within the furnace, which is used as a process variable in a classic feedback regulatory control that manipulates a gas exhausting fan to maintain the process at the desired value. The purpose of this work is to virtualize this residual oxygen analyzer robot, aiming to reduce oxygen variability using artificial intelligence, applying different types of algorithms to build the predictive model, seeking the one with the best accuracy and generalization power based on the indicators of RMSE performance for accuracy and coefficient of determination for generalization power. Decision tree, random forests, XGBoost and LightGBM machine learning algorithms were compared.

Keywords: pulp; lime kiln; virtual sensor; residual oxygen; artificial intelligence; machine learning.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE SÍMBOLOS

ºC	Graus Celsius
t/d	Tonelada por dia
kg/h	quilos por hora
Nm^3/h	Normal metro cúbico por hora
t/h	Tonelada por hora
MJ/ton	Mega joule por tonelada
r.p.m.	Rotações por minuto
Nm^3/t	Normal metro cúbico por tonelada
Pa	Pascal
kg/m^3	Quilos por metro cúbico
m^3/h	Metro cúbico por hora
ppm	parte por milhão
mg/Nm^3	Miligrama por normal metro cúbico
SDCD	Sistema Digital de Controle Distribuído

SUMÁRIO

1 INTRODUÇÃO

A rápida e crescente evolução tecnológica tem causado mudanças no cenário industrial global. As indústrias de celulose também têm sentido o aumento da disputa e dos requisitos no setor, necessitando cada vez mais otimizar o uso dos seus recursos para se manterem competitivas frente ao mercado.

Sendo uma *commodity* de alto valor agregado, a celulose tem tido foco nos últimos anos aos olhos dos grandes investidores, devido ao seu crescente uso na fabricação de produtos de uso diário, tais como fraldas, papeis, cigarros e até mesmo embalagens de alimentos e bebidas que até então eram fabricados usando plástico como constituinte [1].

Buscando atender um mercado exigente do ponto de vista ambiental, bem como reduzir os custos de produção, essas indústrias possuem uma grande parte da sua planta voltada à recuperação de químicos do processo produtivo. Este macroprocesso é conhecido como recuperação e utilidades. Dentro desta grande ilha encontra-se a planta de licor branco, que objetiva a recuperação dos químicos utilizados no cozimento da madeira e a transformação destes para serem reutilizados no processo, se tornando um circuito fechado [2].

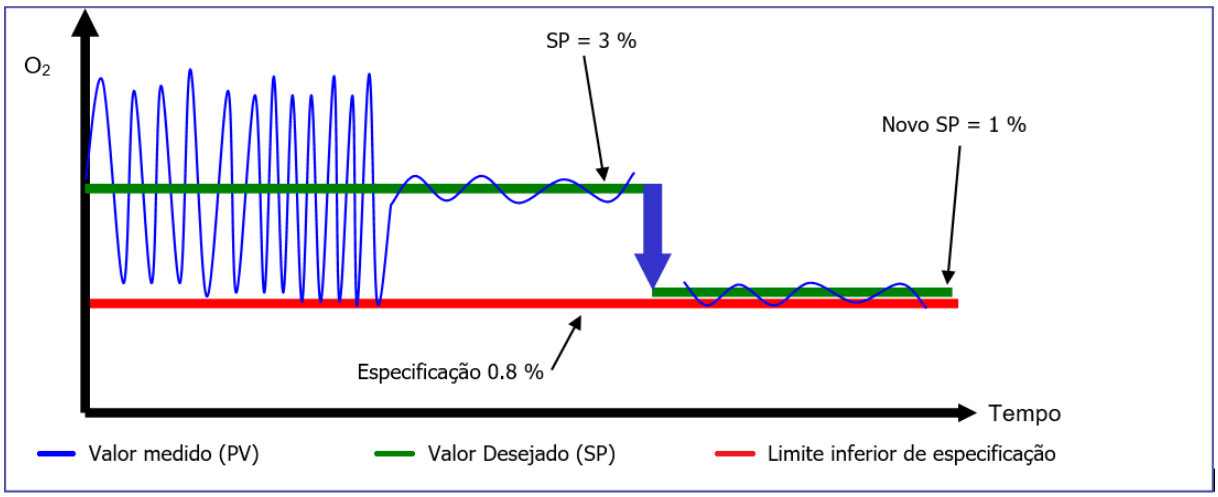
A planta de licor branco é composta por duas áreas, sendo a caustificação a responsável pela conversão do licor verde, proveniente da caldeira de recuperação, em licor branco para ser utilizado no digestor. Para que esta conversão ocorra, se faz necessário a utilização de cal que irá reagir com o licor verde para formar o licor branco. Entretanto por se tratar de uma reação não completa, há a formação de uma lama de cal rica em carbonato de cálcio, o qual pode ser convertido em cal queimada com a adição de energia térmica, por meio da utilização de um forno cilíndrico rotativo [3].

1.1 Sensor virtual de oxigênio residual

Em função dos problemas de disponibilidade, confiabilidade e variabilidade da medição de oxigênio residual do robô analisador, fruto da agressividade do meio que ele se encontra, a equipe operacional costuma trabalhar com um valor desejado de oxigênio residual bem acima do ponto ótimo, por conta da sua variabilidade, ilustrado pela Figura 1. Esse fator acarreta um consumo excessivo de combustíveis para manter a carga térmica dentro do patamar de calcinação da lama, tudo isso ainda é agravado pelo ventilador de tiragem dos gases que, no objetivo de manter o oxigênio no seu valor desejado, acaba promovendo uma maior exaustão

dos gases e como consequência uma maior perda de calor, precisando então aumentar o consumo de combustíveis.

Figura 1 – Redução de variabilidade do oxigênio residual.



Fonte: Próprio autor (2021).

Para resolução deste problema complexo, se viu a necessidade de utilizar técnicas de inteligência artificial para virtualizar esta medição de oxigênio residual, reduzindo sua variabilidade, aumentando a sua disponibilidade já que o sensor virtual pode operar em caso de indisponibilidade do robô, aumentando a robustez do controle e confiabilidade da medição. Com isso é possível operar o forno de cal em condições ótimas, reduzindo o consumo de combustíveis, custo com troca de refratário do forno e até mesmo o custo de manutenção do ventilador de tiragem por conta deste operar em um patamar inferior de rotação.

Sensores virtuais são criados com o objetivo de predizer variáveis de interesse de um processo, podendo substituir, em casos de falha, o sensor real. Seja por conta do custo de sensores físicos, valor de manutenção, dificuldades técnicas para realizar calibração, ou até mesmo necessidade de métodos não convencionais para a medição desta variável [4].

1.2 JUSTIFICATIVA

Atualmente, existem no forno de cal, dois controles clássicos do tipo *feedback* que estão diretamente relacionados, o controle de carga térmica e o controle de oxigênio residual. O primeiro tem o objetivo de manter a energia térmica necessária para promover a reação de calcinação da lama, modulando a vazão de gás natural e mantendo as vazões de hidrogênio e metanol fixadas. Já o controle de oxigênio tem o objetivo de garantir a relação combustível e comburente, evitando que a mistura seja muito rica ou muito pobre, causando altos níveis de monóxido de carbono e esfriamento do forno respectivamente.

O controle atual de oxigênio residual é realizado por um controlador clássico que manipula o ventilador de tiragem dos gases do forno e faz a medição do oxigênio residual por intermédio de um analisador de gases. Por ser um analisador do tipo amostragem, este sofre com uma gama de problemas, desde a necessidade de limpeza automatizada da sonda de coleta a cada 30 minutos, além de problemas com o transporte da amostra como furos na mangueira que possibilitam a entrada de ar falso. Entupimento de filtro também é uma outra questão a ser tratada devido à agressividade do ambiente.

Os problemas do analisador podem acabar causando uma indisponibilidade dessa medição, fazendo com que os operadores desliguem a malha de controle passando a operá-la de forma manual. Contudo, a operação em manual, costuma trabalhar com a referência de velocidade do ventilador de exaustão dos gases em um patamar muito acima do valor otimizado, causando assim a perda de calor do forno pela chaminé e os demais controles de queima acabam por injetar mais combustível para compensar a perda de caloria, elevando assim o custo de um dos maiores consumíveis da fábrica.

A proposta do projeto é investigar o desenvolvimento de um sensor virtual de analisador de oxigênio residual, fazendo uso de algoritmos de Aprendizado de Máquina (AM) supervisionados. Para que posteriormente seja possível predizer o oxigênio residual no forno a cada 10 segundos e permitir ao controlador a correção da velocidade do ventilador de exaustão de forma antecipada (*feedforward*) e, portanto, mais otimizada. Ficando a cargo do atual controle tipo *feedback* realizar pequenas correções remanescentes de eventuais erros do modelo de predição.

Para o problema estudado, serão investigados quatro tipos de algoritmos de AM que serão avaliados em relação ao poder de generalização, acurácia, bem como outras métricas de avaliação. Será aplicada para avaliação dos algoritmos a validação cruzada.

Outro benefício esperado com o sensor virtual é a possibilidade de realizar *cross check* com o sensor real (analisador), possibilitando encontrar defeitos ainda em estágio inicial e informando inconsistências nas variáveis de entrada do modelo, no analisador ou até mesmo indicando um erro do modelo utilizado.

Tal estratégia se faz bastante pertinente pois permite identificar defeitos no sistema antes que este evolua até a falha, provocando perdas maiores no processo, tais como consumo elevado de combustíveis ou até mesmo o entregável do forno fora da especificação ocasionando problemas nas etapas posteriores do processo.

Outro ponto positivo é a possibilidade de se utilizar apenas o sensor virtual em caso de falha do sensor real (analisador) por um tempo longo. Este ponto é especialmente interessante porque a reposição de peças do analisador, possuem um alto custo de importação, podendo reduzir assim níveis de estoque de sobressalentes. Neste cenário não será necessário colocar o controlador em modo manual durante o período de indisponibilidade do analisador.

1.3 OBJETIVOS

1.3.1 Gerais

Almejando reduzir o consumo de gás natural, o foco desse trabalho será utilizar dados do forno de cal da fábrica Veracel Celulose S/A, onde são utilizados algoritmos de AM supervisionado para realizar a tarefa de regressão, de forma que seja possível predizer o volume de oxigênio residual do forno com um nível aceitável de erro. Dentre os algoritmos escolhidos estão a árvore de decisão simples, a floresta aleatória, o *XGBoost* e o *LightGBM*.

A escolha destes algoritmos se deu baseada no tipo do problema e para avaliar a diferença entre a técnica de *bagging* (*random forest*), *gradient boosting* (*XGBoost* e *LightGBM*) e o algoritmo simples de árvore de decisão, que diferente dos últimos três não pratica a técnica de *ensemble*. São utilizadas duas métricas de avaliação do sensor virtual, sendo elas, *Root Mean Squared Error* (*RMSE*) e R^2.

1.3.2 Específicos

A principal questão de pesquisa estudada neste projeto é: "É possível utilizar algoritmos de AM, como descritos anteriormente, para a construção de um sensor virtual analisador de oxigênio residual que permitirá otimizar o processo de forno de cal?". Outra questão tratada é: “Qual algoritmo de AM, dentre os escolhidos neste trabalho, se comportará melhor para predizer o volume de oxigênio residual de um forno?”

Para responder a essas perguntas são utilizados indicadores de desempenho para avaliação da acurácia e do poder de generalização dos modelos obtidos pelos diferentes algoritmos. Com o objetivo de se obter um alto desempenho na estratégia de controle atual, foi estipulado um *RMSE* menor do que 0,25% e para avaliar o poder de generalização do modelo, é utilizado o indicador de desempenho R^2 (coeficiente de determinação) com uma meta superior a 85%.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Forno de cal

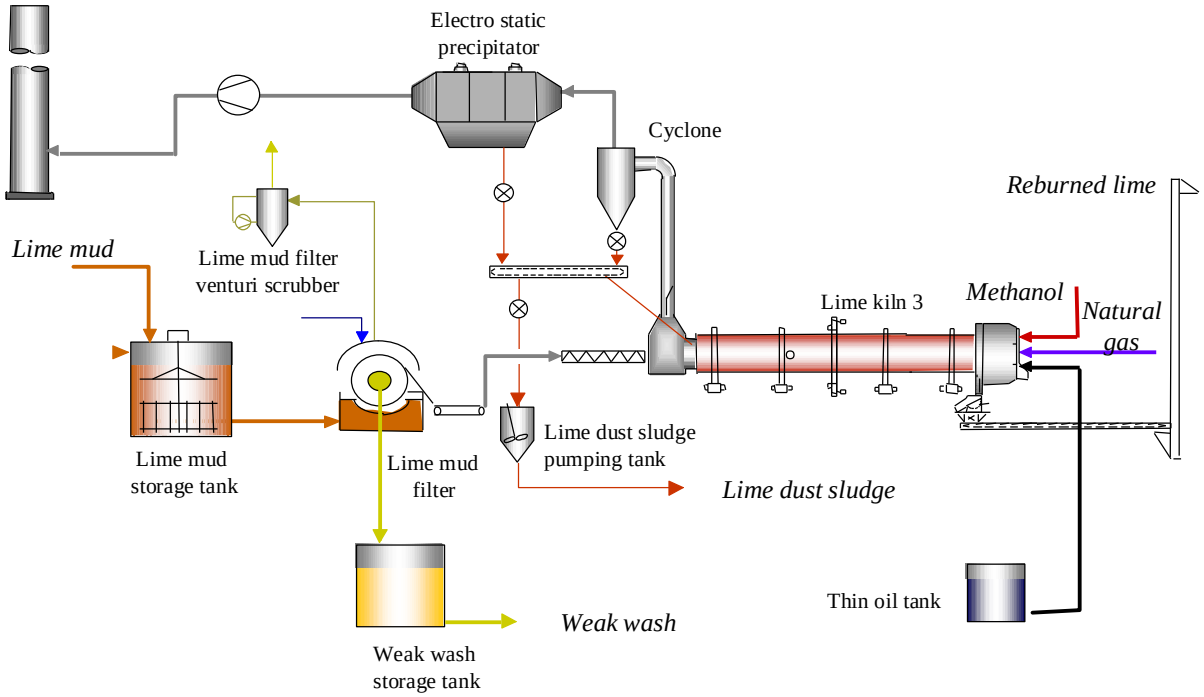
O processo de transformação da lama de cal em cal queimada, acontece na área da fábrica denominada forno de cal, conforme ilustrado na Figura 2. Esta área fornece altas temperaturas, chegando a atingir mais de 650 ºC, para promover a reação endotérmica de calcinação da lama, convertendo-a em cal útil queimada para ser reutilizada no processo [2].

O processo de combustão utilizado na fábrica em questão para se atingir tal temperatura é composto por três diferentes combustíveis queimados simultaneamente, o hidrogênio advindo da planta química como subproduto na produção de químicos utilizados na etapa de branqueamento, o metanol proveniente da etapa de evaporação do licor já utilizado pelo digestor no cozimento da madeira e o gás natural de petróleo comprado de uma companhia externa à fábrica.

A operação do forno de cal é desafiadora, visto que existem muitas variáveis que impactam no processo de conversão da cal. Para reduzir o custo de produção, utiliza-se ao máximo os combustíveis gerados do processo, completando com o gás natural para atingimento da carga térmica necessária. Outro grande desafio é a qualidade dos combustíveis, visto que dois destes são subprodutos e têm bastante variabilidade no poder calorífico por conta da falta de pureza [6].

Por ser um processo de combustão, o oxigênio residual é uma das principais variáveis a serem controladas para otimizar a relação comburente/combustível, já que uma mistura muito rica em combustível gerará uma queima incompleta e bastante monóxido de carbono, o qual causa risco de explosão no equipamento de filtragem das cinzas. Por outro lado, uma queima pobre em combustível tende a resfriar o forno e por consequência desfavorecer a reação de calcinação da lama.

Figura 2 – Visão geral forno de cal.



Fonte: ANDRITZ – *AG GMBH* (2007).

2.2 Regressão e principais algoritmos

Durante uma investigação de cunho científico hipóteses são formuladas a respeito dos relacionamentos entre variáveis de entrada ou independentes (***x***) que possam explicar parcialmente ou até mesmo totalmente uma variável de saída ou dependente (*y*), possibilitando desta forma a criação de um modelo de regressão para realização de predições de *y* baseado em valores conhecidos de *x*. A linearidade do modelo se dá quando as variáveis de entrada e saída possuem um relacionamento linear ou aproximadamente linear entre elas, quando as variáveis não possuem esse tipo de relacionamento se considera um modelo não linear [7]. Para a construção de um modelo de regressão linear se faz necessário que todas as variáveis de entrada sejam quantitativas.

Para medição da acurácia de cada modelo são utilizadas algumas medidas, como o coeficiente de determinação (R^2), o erro médio absoluto (*MAE*), a raiz quadrada do erro médio quadrático (*RMSE*) e o erro médio quadrático (*MSE*), conforme as equações a seguir. Essas medidas apontam a disparidade entre a saída real e a saída predita possibilitando a comparação dos indicadores entre os modelos obtidos de forma quantitativa. O *RMSE* é a medida mais rigorosa em penalizar de forma mais severa os valores discrepantes em relação à média.

$$R^2 = \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (01)$$

$$MAE = \sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{n} \quad (02)$$

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(y_i - \hat{y}_i)^2}{n}} \quad (03)$$

$$MSE = \sum_{i=1}^{n} \frac{(y_i - \hat{y}_i)^2}{n} \quad (04)$$

sendo *n* é o número de instâncias, y_i é o *i*-ésimo valor real, $\hat{y}_i$ é a *i*-ésima predição e $\bar{y}$ é o valor médio de y.

D*ecision tree* ou árvores de decisão são algoritmos de aprendizado de máquina para inferência de regras de decisão simples que recebem um vetor de *features* (***x***) ou valores como entrada e tem como saída uma decisão única (*y*). Sendo um algoritmo de aprendizado de máquina supervisionado bastante útil para regressão e classificação e amplamente utilizado em abordagens de modelagem preditiva usadas em estatística, mineração de dados e aprendizado de máquina. Quando aplicado em problemas de regressão, a estrutura desse algoritmo baseia-se em folhas representando rótulos de classe, e os ramos representando conjunções de características que levam a esses rótulos de classe [8].

Árvores de regressão são árvores de decisão em que as variáveis de entrada podem assumir valores contínuos e sendo um dos algoritmos de aprendizado de máquina mais utilizado e conhecido por conta da sua, simplicidade, inteligibilidade e facilidade de implementação, contudo há uma grande probabilidade de ocorrências de *overfitting* ou sobre ajuste. A construção dessas árvores é feita por algoritmos que funcionam de cima para baixo, escolhendo em cada etapa a variável que divide da melhor forma o conjunto de itens. São utilizados diferentes algoritmos com métricas diferentes para mensuração do melhor modelo, medindo a homogeneidade da variável dependente dentro dos subconjuntos. Podem ser citados como métodos utilizados para construção das árvores a impureza de Gini e o ganho de informação, sendo aplicados em cada um dos subconjuntos candidatos combinando os resultados combinados e fornecendo uma medida da qualidade desta divisão [8].

A *random forest* ou floresta aleatória assim como a árvore de decisão, é um algoritmo de aprendizado de máquina supervisionado usado para classificação ou regressão possuindo a capacidade de combinar a simpleza das árvores de decisão com aleatoriedade para melhorar a precisão, robustez e flexibilidade. Tendo em vista que a chance de ocorrência do *overfitting* nas árvores de decisão cresce na medida em que o tamanho e a complexidade da árvore aumentam, podendo reduzir o poder de generalização do modelo quando o algoritmo for submetido ao ambiente de produção [9].

O algoritmo tem seu funcionamento dado pela criação de múltiplas árvores realizando posteriormente um método de *ensemble* ou junção, das diferentes variáveis e conjuntos de dados já contidos no *data frame* ou base de dados de forma aleatória, ao invés da escolha partir do cálculo de impureza como é feito em uma árvore de decisão simples, reduzindo assim a ocorrência do *overfitting*.

A definição do número de árvores que o algoritmo criará, é feita por meio de um hiperparâmetro denominado número de estimadores, onde o resultado de cada modelo é combinado para a extração de um único valor final (*bagging*). Numa comparação entre os algoritmos de floresta aleatória e árvore de decisão simples, a *random forest* exige maior poder computacional, diretamente proporcional ao número de árvores definido, porém proferindo robustez, no que diz *respeito ao overfitting*, e normalmente maior acurácia [10].

XGBoost deriva de *eXtreme Gradient Boosting*, representando uma classe de algoritmos baseada em árvores de decisão com aumento do gradiente (*Gradient Boosting*). O aumento de gradiente quer dizer que o algoritmo faz uso do artifício de *Gradient Descent* para encontrar o ponto de mínima perda, permitindo que novos modelos sejam adicionados. *Gradient Boosting* é a capacidade de combinar resultados de diversos classificadores com vieses fracos, normalmente *Decision Trees*, que são combinados com o objetivo de formar algo como um "comitê de decisão robusto" [11].

Por possuir uma diversidade de hiperparâmetros passíveis de sintonia, este algoritmo se torna altamente adaptável, permitindo desta forma o ajuste adequado para solução de diversos tipos de problemas. O seu funcionamento se dá com a criação sequencial de árvores, tendo como base resíduos anteriores (*boosting*) e como passo seguinte realiza a derivada do erro atual do conjunto [12].

Outro algoritmo baseado em árvore de decisão é o *LightGBM*. Este tem como vantagem sobre o *XGBoost* a capacidade de promover maior velocidade de treinamento, além de uma eficiência superior somado a um menor uso de recurso computacional, com melhor precisão,

suporte de aprendizagem paralela e *GPU* e ainda tem maior capacidade para lidar com grandes volumes de dados de múltiplas fontes simultaneamente.

Duas diferentes estratégias são utilizadas para computar as árvores, sendo a primeira delas a *level-wise*, utilizada pelos algoritmos *XGBoost* e *random forests*, e a *leaf-wise* utilizada para compor o *LightGBM*. Na estratégia *level-wise* aumenta-se a árvore nível a nível, com cada nó dividindo os dados e priorizando os nós mais próximos da raiz da árvore. Já com a estratégia *leaf-wise* o crescimento da árvore é dado pela divisão dos dados nos nós com a maior mudança de perda. Comparando as duas estratégias, tem-se em geral, um melhor desempenho com *level-wise* para conjuntos de dados menores, visto que a *leaf-wise* tende ao ajuste excessivo o que pode levar ao *overfitting*. Para conjunto de dados maiores, o cenário *leaf-wise* tende a se sobressair, por causa da sua maior velocidade de execução, fruto de uma melhor otimização interna quando comparado ao *level-wise* [13].

Outro ponto importante do algoritmo *LightGBM* é a aproximação da divisão realizada por meio da construção de histogramas dos atributos, assim não é necessário realizar a avaliação de cada valor único dos atributos para cálculo da divisão, mas apenas os *bins* do histograma, os quais são limitados, tornando essa abordagem mais eficiente para grandes volumes de dados sem afetar de forma negativa a precisão.

3 TRABALHOS RELACIONADOS

Com um grande acervo de trabalhos correlatos disponíveis nas bases de dados das academias, foi observado a aplicação de algoritmos de aprendizado de máquina para construção de sensores virtuais em diferentes processos industriais. Dentre os trabalhos encontrados, dois se destacaram sendo mais fortemente relacionados com a virtualização do robô analisador de oxigênio residual em forno de cal.

Lin *et al* (2007) com o objetivo de criar um sensor virtual de cal livre para forno de cimento, utilizou um *template* baseado em um ou mais variáveis chave de processo para lidar com dados ausentes, realizando em seguida análise de componentes principais (*PCA*) para redução de dimensionalidade. Ao final utiliza de técnicas robustas de regressão para encontrar um modelo de inferência. Um modelo de mínimos quadrados parciais dinâmicos (*DPLS*) é implementado para resolver o problema da autocorrelação nos dados do processo e, assim, fornece uma estimativa mais suave do que usar um modelo de regressão estático. Atingindo um nível de ótimo de significância de 95,5% através dos testes estatísticos T^2 e Q.

Pan *et al* (2021) construiu um sensor virtual de oxigênio baseado em redes de memória de curto e longo prazo (*LSTM*) com o intuito de predizer o volume de oxigênio de caldeiras. Inicialmente o modelo *LSTM* foi construído com o *Keras deep learning framework*. Foi utilizado também algoritmos baseado em máquina de vetores de suporte (*SVM*) para a construção dos mesmos sensores. Houve um aumento significativo na acurácia do modelo após seleção apropriada de hiperparâmetros pós testes práticos e foi utilizado o indicador de desempenho *RMSE* para comparação entre o algoritmo *LSTM* e *GA-SVM* construídos, atingindo um erro de 4,51% menor utilizando *LSTM*.

4 MATERIAIS E MÉTODOS

Com o intuito de desenvolver um sensor virtual de um robô analisador de oxigênio residual para ser utilizado em uma malha de controle e assim otimizar o consumo de gás natural do forno de cal, é necessário o emprego de algumas técnicas de aprendizado de máquina. Para que o processo de aprendizado tenha seus resultados maximizados é preciso seguir algumas premissas desde a extração, transformação e carga dos dados que servirão de base para a construção de modelo.

A extração dos dados é realizada por meio do sistema de gestão de informações da planta (*PIMS*) em arquivos de valores separados por vírgula, uma extensão amplamente compatível com os *softwares* de mercado existentes. Os dados são coletados dos sensores em campo e armazenados no banco de dados sistema de informação, normalmente, a cada 5 segundos, podendo variar de acordo com o tempo de resposta de cada variável.

O tempo de varredura escolhido para a extração dos dados e montagem dos *datasets* foi definido como média de 1 minuto, contendo um período de 01/01/2022 a 01/06/2022. Este período foi definido, em consenso com a equipe de engenheiros químicos e time operacional, que abrange boa parte dos cenários possíveis da dinâmica do processo do forno, contribuindo desta forma para um melhor poder de generalização e acurácia dos algoritmos utilizados.

A definição das variáveis a serem utilizadas para construção do *dataset* é baseada em entrevista com os engenheiros de processo somados aos operadores de todos os cinco turnos que trabalham todos os dias do ano, vinte e quatro horas por dia. Esta equipe multidisciplinar com conhecimento tanto teórico quanto prático, possuem a capacidade de indicar variáveis que possuem algum impacto sobre o atributo a ser predito, neste caso sendo o oxigênio residual. Ao realizar tal levantamento, formou-se o *dataset* contendo 30 atributos, sendo um deles a variável a ser predita (3501QC019.MEAS) e os demais variáveis correlatas, conforme mostrado na Tabela 1.

Tabela 1 – Variáveis utilizadas.

Tag	Descrição	Unidade	Mínimo	Máximo
3502FI353.PNT	CÁLCULO PROD FORNO	t/d	0	800
3501FC144.MEAS	HIS PARA QUEIMADOR	kg/h	0	500
3501FC288.MEAS	FLUXO GNP	Nm³/h	0	10000
3501FC176.MEAS	MET P/ QUEIMADOR	t/h	0	2
3501FI216.PNT	CALOR P/ TON CAL	MJ/ton	0	10000
3501TI136.PNT	CAMADA ZONA QUEIM	°C	0	1000
3501TI135.PNT	FLUXO DE GASES	°C	0	1000
3501CONS_ESP.RO04	CONSUMO ESP GAS	Nm³/t	0	500
3501QC019.MEAS	O2 RES SEC LAMA	%	0	100
3501TC013.MEAS	SECADOR DE LAMA	°C	0	1000
3501TC027.MEAS	SECADOR DE LAMA	°C	0	1000
3501PI012.PNT	SECADOR DE LAMA	Pa	-10000	10000
350142002VEL.PNT	ROSCA 1 ALIM DE LMU	%	0	100
3501PI016.PNT	GASES EXAUST P/ PE	Pa	-10000	10000
3501TC017.MEAS	GASES EXAUST P/ PE	°C	0	500
350142041VEL.PNT	VENTIL. AR IND FORNO	%	0	100
3501QM23001CONS	CONSIST. ENT FORNO	%	0	100
3502DC332.MEAS	LMU ALIM FORNO	kg/m³	0	2000
3502FC331.MEAS	LMU P/ FILTRO LAMA	m³/h	0	500
3501QI121.PNT	NOX CHAMINÉ	ppm	0	1000
3501QI120.PNT	TRS CHAMINÉ	ppm	0	1000
3501QI119.PNT	SO2-CHAMINÉ	ppm	0	1000
3501QI118.PNT	PART CHAMINÉ	mg/Nm³	0	1000
3501QI117.PNT	CO-CHAMINÉ	ppm	0	1000
3501QI116.PNT	O2-CHAMINÉ	%	0	100
3502LI311.PNT	TQ ESTOC LMU	%	0	100
3501LI085.PNT	SILO CAL QUEIMADA	%	0	100
3501LI084.PNT	SILO CAL COMPRADA	%	0	100
3502FI187.PNT	CAL RECUP P/ SLAKER	t/h	0	100
3502FI176.PNT	CAL VIRGEM P/ SLAKER	t/h	0	100

Fonte: Próprio autor (2022).

Uma vez exportado os dados no formato de arquivo valor separado por vírgula (*.csv*) com as variáveis organizadas em colunas e as instâncias como tuplas, é necessário tratar os dados para serem utilizados na próxima etapa, conhecida como extração de padrão. O tratamento desses dados será realizado com algumas técnicas de filtragem interpretados em linguagem *python* no ambiente *Google Colab notebooks*.

Os dados foram inspecionados graficamente utilizando *boxplots* para a verificação de valores destoantes ou *outliers*, primeiro e terceiro quartil e a faixa interquartil (FIQ). As instâncias que continham valores destoantes foram removidas, segundo as equações (05), (06) e (07) em cada atributo. Outras instâncias removidas juntamente com os valores destoantes foram as que continham valores inconsistentes, dados faltantes ou dados com sinal congelado oriundo de falha de comunicação da base de dados com o sistema de automação.

Uma vez removido os *outliers*, foram novamente traçados os gráficos *boxplots* para avaliação do resultado da exclusão, além da comparação do antes e depois das medidas de curtose (*Kurtosis*) e momentos de obliquidade (*Skewness*), para avaliar o quanto os dados se encontram distribuídos em relação a curva normal. Utilizou-se também o recurso de mapa de

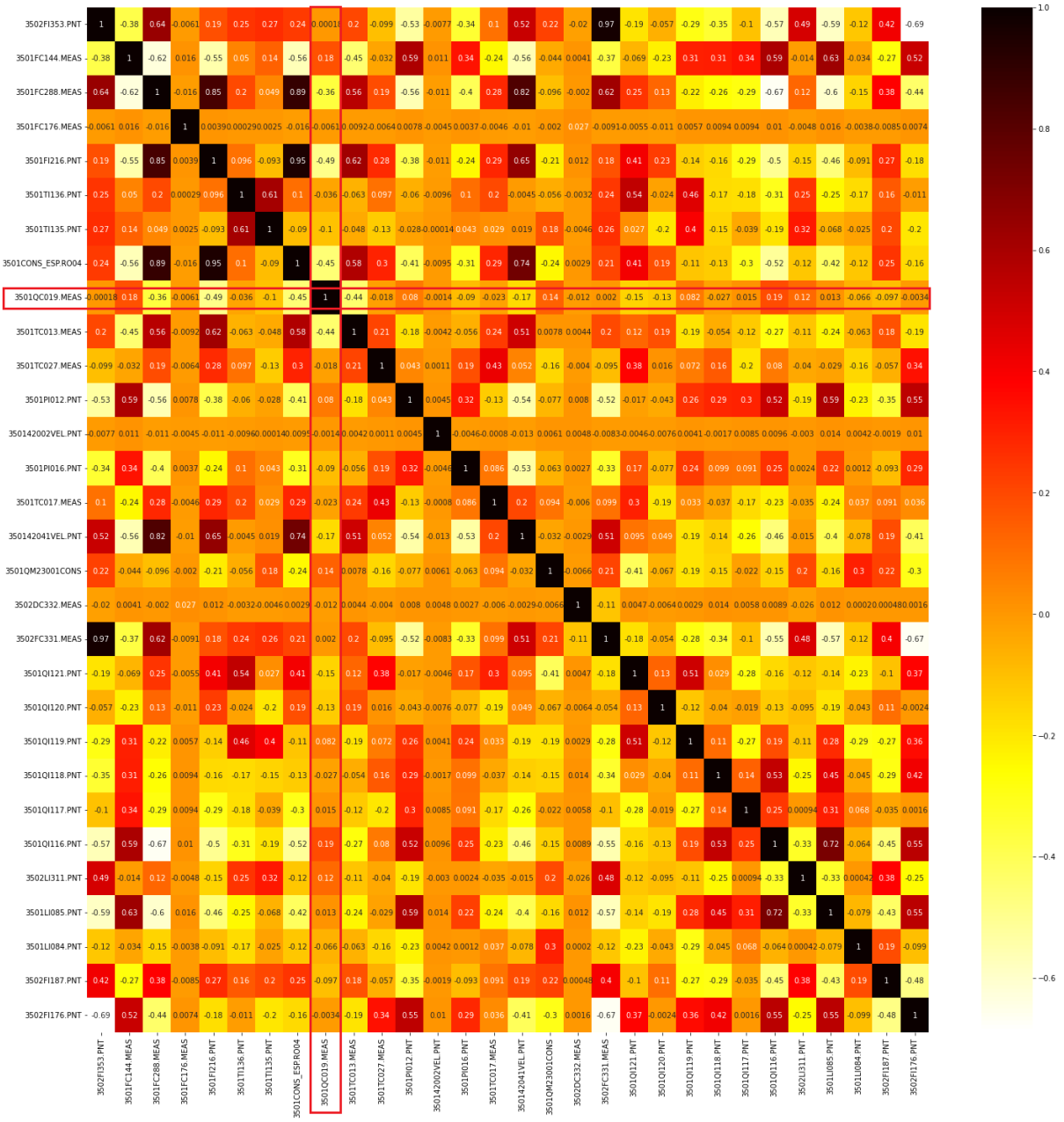
calor, conforme ilustrado na Figura 3, para a validação de correlação entre a variável a ser predita e os demais atributos, podendo assim confirmar a correta escolha das variáveis baseado na experiência dos engenheiros de processo e operadores da fábrica.

$$FIQ = Q3 - Q1 \quad (05)$$

$$Outlier\ inferior = Q1 - 1{,}5 * FIQ \quad (06)$$

$$Outlier\ superior = Q3 + 1{,}5 * FIQ \quad (07)$$

Figura 3 – Mapa de calor correlacionando variáveis do *dataset*.



Fonte: Próprio autor (2022).

Com os dados já extraídos da fábrica, tratados e pré-processados, se faz possível a aplicação de métodos de aprendizado de máquina para extrair padrões de comportamento do atributo a ser virtualizado (predito). Para isto, foram escolhidos os algoritmos de árvore de decisão, florestas aleatórias, *XGBoost* e *LightGBM*. Foram levantados os indicadores *Mean Absolute Error* (*MAE*), *Mean Squared Error* (*MSE*) e *Root Mean Squared Error* (*RMSE*) para cada algoritmo, com o intuito de avaliar e comparar a acurácia de cada modelo. Para avaliação do poder de generalização de cada algoritmo, foi utilizado coeficientes de determinação (R^2).

Para iniciar a aplicação dos algoritmos, foi necessário segmentar os dados em dois *datasets*, sendo o primeiro contendo somente a variável de saída (y) e o segundo contendo as variáveis de entrada ($\boldsymbol{x}$). Uma vez segmentando os dados, os *datasets* foram novamente partidos em dois conjuntos através da utilização do parâmetro *test_size*, contido no algoritmo *train_test_split* da biblioteca *sklearn*. O valor do parâmetro foi definido como 0,5, ou seja, metade dos dados fará parte do conjunto de treinamento e a outra metade do conjunto de teste.

Todos os algoritmos utilizados neste trabalho tiveram o mesmo método de ajuste do modelo, sendo realizado seguindo os passos desde a extração dos dados, limpeza e filtragem de valores destoantes, faltantes e inconsistentes, passando pela formação de *dataset*, separação dos conjuntos de treinamento e teste, posteriormente com os indicadores de poder de generalização e acurácia em mãos é realizado a sintonia dos hiperparâmetros baseado em tentativa e erro, buscando o menor erro do indicador *RMSE* nos dados de treino.

Normalmente, para o tipo de problemática proposta neste trabalho, o algoritmo *XGBoost* costuma ter um desempenho prático superior ao algoritmo de florestas aleatórias, que por sua vez costuma ser superior ao algoritmo de árvore de decisão simples. Já o *LightGBM* costuma possuir resultado similar ao *XGBoost*, porém com uma velocidade de processamento superior. Estes algoritmos foram os escolhidos para terem suas performances evidenciando assim o melhor algoritmo a ser utilizado para solução da virtualização de um sensor virtual de oxigênio residual em um forno de cal.

5 RESULTADOS E DISCUSSÕES

5.1 Sintonia dos parâmetros

Começando com o algoritmo árvore de decisão para realizar a virtualização do oxigênio residual, foi utilizado o algoritmo *Decision Tree Regressor* da biblioteca *sklearn*, utilizando os hiperparâmetros *min_samples_leaf* e *max_depth* e tendo o *random_state* fixado num valor único para permitir e facilitar a reprodutibilidade dos testes de sintonia realizados.

O *min_samples_leaf* define o valor mínimo de amostras para a formação de um nó folha, o objetivo do seu ajuste é mitigar ou até mesmo evitar a formação de nós folhas com *outliers*, para que não haja sobreajuste (*overfitting*). Já o parâmetro *max_depth* é utilizado também a fim de mitigar ou evitar o *overfitting*, porém o seu funcionamento se dá através da limitação do crescimento de ramos da árvore, também conhecido como nível de profundidade.

Sintonizando ambos os hiperparâmetros para uma árvore de decisão simples, fixando um deles e variando o outro e avaliando as métricas, foram obtidos os resultados conforme a Tabela 2. Mesmo com a melhor sintonia encontrada dos parâmetros, ainda não foi possível atingir a meta estabelecida pelo trabalho de 0,25% de *RMSE*, fazendo assim necessário a implementação dos próximos algoritmos.

Tabela 2 – Ajuste dos hiperparâmetros *max_depth* e *min_samples_leaf* em árvore de decisão simples dados de treino.

Algoritmo	**Hiperparâmetro**	**Valor**	**Métricas de acurácia**				**Comentários**
			MAE	***MSE***	***RMSE***	***R²***	
Decision Tree	*min_samples_leaf* *max_depth*	1 5	0,32723	0,17739	0,42118	0,52634	Sub-ajuste
	min_samples_leaf *max_depth*	1 *none*	0,00000	0,00000	0,00000	1,00000	Sobreajuste
	min_samples_leaf *max_depth*	250 *none*	0,24280	0,09955	0,31552	0,73418	Ajuste bom

Fonte: Próprio autor (2022).

Seguindo para o próximo algoritmo com o intuito de reduzir ainda mais o erro segundo as métricas adotadas, foi escolhido o algoritmo de florestas aleatórias por utilizar múltiplas árvores de decisão simples e por ser relativamente robusto à sobreajuste. Também foi utilizada a biblioteca *Sklearn* importando o algoritmo *Random Forest Regressor* e realizada a sintonia dos parâmetros *max_depth* assim como utilizado no algoritmo de árvore de decisão simples com a adição do parâmetro *n_estimators*, o qual se trata do número de árvores que estará contida na floresta. Os resultados podem ser observados conforme Tabela 3.

Tabela 3 – Ajuste dos hiperparâmetros *n_estimators* e *max_depth* em florestas aleatórias dados de treino.

Algoritmo	Hiperparâmetro	Valor	Métricas de acurácia				Comentários
			MAE	***MSE***	***RMSE***	***R²***	
Random Forest	*n_estimators* *max_depth*	100 5	0,26924	0,11811	0,34367	0,68463	Sub-ajuste
	n_estimators *max_depth*	100 *none*	0,04494	0,00361	0,06008	0,99036	Sobreajuste
	n_estimators *max_depth*	300 7	0,20355	0,06829	0,26133	0,81765	Ajuste bom

Fonte: Próprio autor (2022).

A sintonia dos hiperparâmetros foi realizada através da tentativa e erro, variando somente um por vez e fixando os demais até se atingir o menor erro possível (segundo o *RMSE*) para aquele parâmetro nos dados de treino. Apesar da melhora dos indicadores de acurácia em relação ao algoritmo de árvore de decisão simples, o indicador *RMSE* ainda não atingiu a meta estabelecida pelo projeto, sendo necessário a aplicação de outros algoritmos.

O próximo algoritmo a ser utilizado é o *XGBoost* que diferente da árvore de decisão simples e das florestas aleatórias, utiliza uma técnica conhecida como *boosting*, ou treinamentos sucessíveis baseados na derivada do erro com o objetivo de reduzir severamente o erro do modelo. Novamente utilizada a biblioteca *Sklearn*, importado o algoritmo *XGBoost Regressor* e sintonizados os parâmetros *n_estimators*, *max_depth* e *learning_rate*.

O parâmetro *n_estimators* para este algoritmo corresponde ao número de estágios de treinamentos sucessíveis e o *learning_rate* à taxa de aprendizado ou a redução da contribuição de cada árvore. Foi possível também atingir a meta estipulada pelo projeto de ao utilizar o *XGBoost* conforme a Tabela 4, a melhora da métrica *RMSE* e *R²* em relação ao algoritmo de florestas aleatórias e árvore de decisão simples foi bastante significativa.

Tabela 4 – Ajuste dos hiperparâmetros *n_estimators*, *max_depth* e *learning_rate* do *XGBoost* dados de treino.

Modelo	Hiperparâmetro	Valor	Métricas de acurácia				Comentários
			MAE	***MSE***	***RMSE***	***R²***	
XGBoost	*n_estimators* *max_depth* *learning_rate*	100 3 0,1	0,24482	0,11101	0,33318	0,86139	Sintonia padrão
	n_estimators *max_depth* *learning_rate*	375 10 0,05	0,04277	0,00327	0,05716	0,99128	Sobreajuste
	n_estimators *max_depth* *learning_rate*	50 3 0,05	0,32213	0,16511	0,40633	0,55914	Sub-ajuste
	n_estimators *max_depth* *learning_rate*	200 5 0,05	0,15229	0,03833	0,19577	0,89766	Ajuste bom

Fonte: Próprio autor (2022).

Com o objetivo de comparar as métricas de acurácia dentre os algoritmos utilizados e tentar reduzir ainda mais o erro, foi utilizado também o algoritmo *LightGBM Regressor* da biblioteca *LightGBM*, sintonizando os parâmetros *n_estimators*, *max_depth*, *learning_rate* e *num_leaves*, sendo este último hiperparâmetro referente ao número máximo de folhas da árvore. As métricas de acurácia obtidas sobre a utilização deste algoritmo se encontram descritos na Tabela 5. O algoritmo *LightGBM* obteve um resultado ainda superior em comparação com o algoritmo *XGBoost* tanto na avaliação do indicador *RMSE* quanto no coeficiente de determinação (R^2).

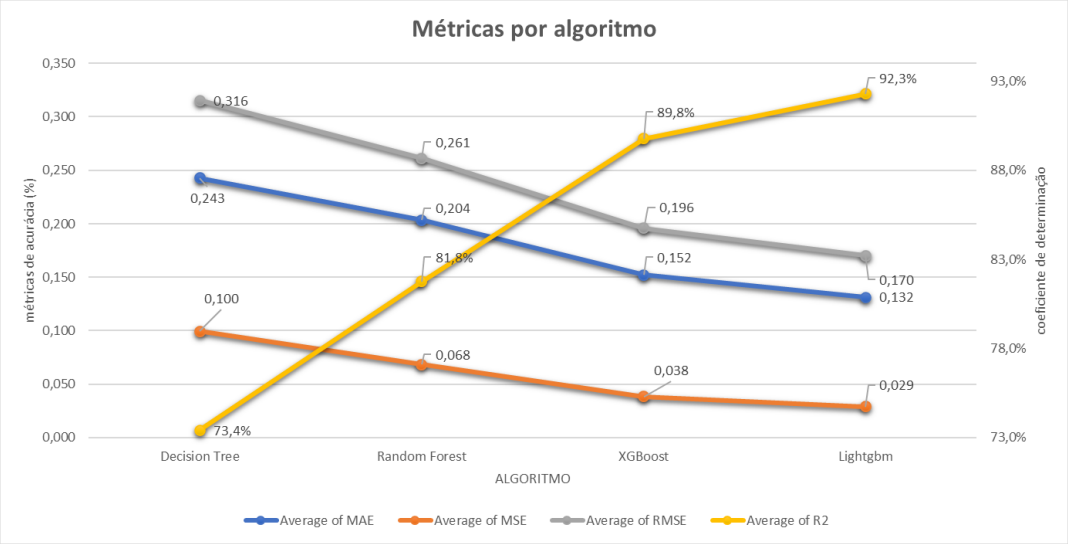
Tabela 5 – Ajuste dos hiperparâmetros *n_estimators*, *max_depth*, *learning_rate* e *num_leaves* do *LightGBM* dados de treino.

Modelo	**Hiperparâmetro**	**Valor**	**Métricas de acurácia**				**Comentários**
			MAE	***MSE***	***RMSE***	***R²***	
LightGBM	*num_leaves*	31	0,13716	0,03106	0,17623	0,91707	Sintonia padrão
	learning_rate	0,1					
	n_estimators	100					
	max_depth	*none*					
	num_leaves	100	0,02180	0,00088	0,02969	0,99765	Sobreajuste
	learning_rate	0,15					
	n_estimators	500					
	max_depth	15					
	num_leaves	2	0,27211	0,11728	0,34246	0,68684	Sub-ajuste
	learning_rate	0,15					
	n_estimators	10					
	max_depth	5					
	num_leaves	15	0,13166	0,02891	0,17003	0,92281	Ajuste bom
	learning_rate	0,15					
	n_estimators	250					
	max_depth	4					

Fonte: Próprio autor (2022).

É possível comparar graficamente os resultados das métricas de acurácia e poder de generalização por algoritmo segundo a Figura 4. Percebendo o aumento gradual do R^2, concomitantemente com a redução do erro em cada indicador de acurácia. A partir do terceiro algoritmo utilizado, já foi possível atingir a meta estipulada pelo projeto de 0,25% de erro segundo o indicador *RMSE* e R^2 superior a 85%, na variável de oxigênio residual virtualizada.

Figura 4 – Métricas de acurácia por algoritmos em dados de treino.



Fonte: Próprio autor (2022).

5.2 Validação com algoritmo *K-fold*

Objetivando avaliar o poder de generalização de cada modelo gerado pelos diferentes algoritmos utilizados sob um mesmo conjunto de dados (*dataset*), foi utilizado o método *K-fold* de validação cruzada, fazendo o uso do algoritmo *K-Fold* da biblioteca *Sklearn*, sobre a respectiva parte dos dados não utilizada para treinamento, o conjunto de teste. Mitigando desta forma a influência dos dados já vistos durante o treinamento dos modelos, que poderiam impactar na avaliação do real poder de generalização.

Os hiperparâmetros ajustados no processo de validação cruzada foram *shuffle*, *n_split* e *random_state*. O parâmetro *shuffle* tem o objetivo de embaralhamento dos *folds*, já o *n_split* remete a quantidade de *folds* que serão utilizados e o *random_state* remete ao estado aleatório utilizado, com o objetivo de permitir reprodutibilidade do experimento. Ajustando o número de *folds*, mantendo o parâmetro *shuffle* em verdadeiro e fixando um valor de *random_state*, obtiveram-se os resultados de cada algoritmo, conforme ilustrados por Tabelas 6, 7, 8 e 9.

Tabela 6 – Validação cruzada *Decision Tree* em dados de teste.

Validação Cruzada *K-fold*		*Decision Tree*								
Hiperparâmetro		*n_splits*	*random_state*	*shuffle*	*n_splits*	*random_state*	*shuffle*	*n_splits*	*random_state*	*shuffle*
Valor		3	7	1	5	7	1	10	7	1
Métricas de Acurácia	Média		0,67478			0,68592			0,70413	
	Desvio Padrão		0,00747			0,01381			0,01588	
	R^2 (1)		0,68414			0,70870			0,72519	
	R^2 (2)		0,67437			0,67060			0,72590	
	R^2 (3)		0,66584			0,69431			0,71324	
	R^2 (4)		-			0,67648			0,67353	
	R^2 (5)		-			0,67953			0,70624	
	R^2 (6)		-			-			0,71393	
	R^2 (7)		-			-			0,70263	
	R^2 (8)		-			-			0,68879	
	R^2 (9)		-			-			0,68844	
	R^2 (10)		-			-			0,70340	

Fonte: Próprio autor (2022).

Tabela 7 – Validação cruzada *Random Forest* em dados de teste.

Validação Cruzada *K-fold*		*Random Forest*								
Hiperparâmetro		*n_splits*	*random_state*	*shuffle*	*n_splits*	*random_state*	*shuffle*	*n_splits*	*random_state*	*shuffle*
Valor		3	7	1	5	7	1	10	7	1
Métricas de Acurácia	Média		0,81422			0,81387			0,81141	
	Desvio Padrão		0,00679			0,00630			0,00760	
	R^2 (1)		0,82242			0,82317			0,81142	
	R^2 (2)		0,81443			0,81593			0,82676	
	R^2 (3)		0,80580			0,81469			0,81805	
	R^2 (4)		-			0,80371			0,81001	
	R^2 (5)		-			0,81187			0,81512	
	R^2 (6)		-			-			0,80888	
	R^2 (7)		-			-			0,81030	
	R^2 (8)		-			-			0,79559	
	R^2 (9)		-			-			0,80647	
	R^2 (10)		-			-			0,81156	

Fonte: Próprio autor (2022).

Tabela 8 – Validação cruzada *XGBoost* em dados de teste.

Validação Cruzada *K-fold*		*XGBoost*								
Hiperparâmetro		*n_splits*	*random_state*	*shuffle*	*n_splits*	*random_state*	*shuffle*	*n_splits*	*random_state*	*shuffle*
Valor		3	7	1	5	7	1	10	7	1
Métricas de Acurácia	Média		0,87901			0,88020			0,88118	
	Desvio Padrão		0,00310			0,00550			0,00501	
	R^2 (1)		0,88161			0,88870			0,88769	
	R^2 (2)		0,88078			0,87415			0,88872	
	R^2 (3)		0,87465			0,88407			0,87506	
	R^2 (4)		-			0,87509			0,87920	
	R^2 (5)		-			0,87901			0,88459	
	R^2 (6)		-			-			0,88511	
	R^2 (7)		-			-			0,87787	
	R^2 (8)		-			-			0,87285	
	R^2 (9)		-			-			0,87947	
	R^2 (10)		-			-			0,88124	

Fonte: Próprio autor (2022).

Tabela 9 – Validação cruzada *LightGBM* em dados de teste.

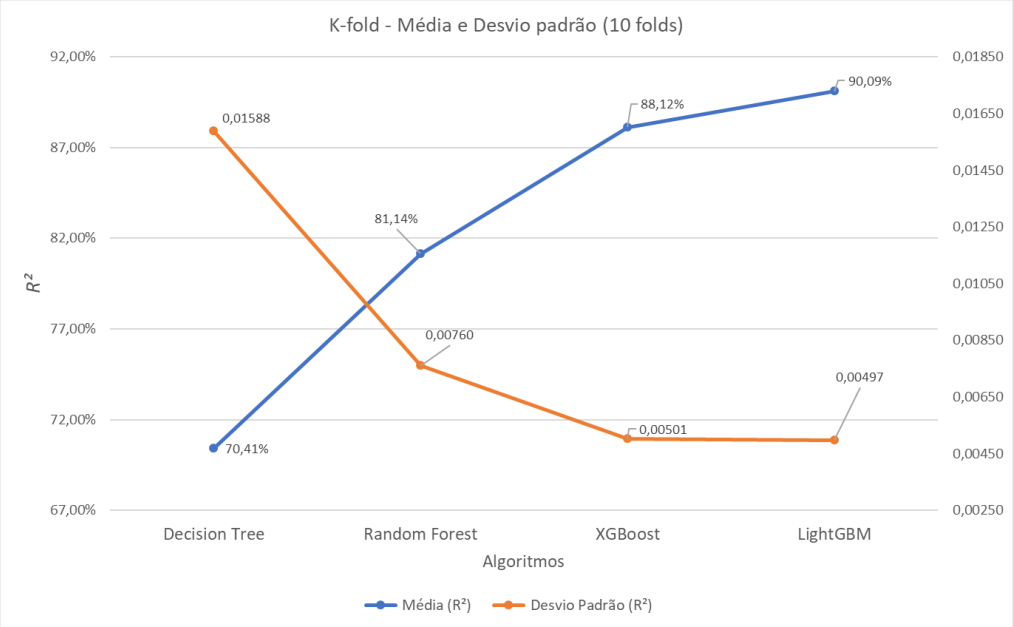
Validação Cruzada *K-fold*		*LightGBM*								
Hiperparâmetro		*n_splits*	*random_state*	*shuffle*	*n_splits*	*random_state*	*shuffle*	*n_splits*	*random_state*	*shuffle*
Valor		3	7	1	5	7	1	10	7	1
Métricas de Acurácia	Média		0,89804			0,89997			0,90090	
	Desvio Padrão		0,00270			0,00419			0,00497	
	R^2 (1)		0,90013			0,90603			0,90609	
	R^2 (2)		0,89975			0,89548			0,90880	
	R^2 (3)		0,89423			0,90390			0,89530	
	R^2 (4)		-			0,89775			0,89697	
	R^2 (5)		-			0,89672			0,90475	
	R^2 (6)		-			-			0,90606	
	R^2 (7)		-			-			0,90244	
	R^2 (8)		-			-			0,89533	
	R^2 (9)		-			-			0,89684	
	R^2 (10)		-			-			0,89642	

Fonte: Próprio autor (2022).

Uma vez obtidos os resultados das validações cruzadas com 3, 5 e 10 *folds*, incialmente, todos os modelos árvore de decisão e floresta aleatória não conseguiram atingir a meta estipulada no projeto em nenhuma das validações (3, 5 e 10 *folds*), apesar do desvio padrão de ambos terem sido consideravelmente baixos. Também é possível identificar a mudança de patamar do valor médio do R^2 ao passar a utilizar o algoritmo de florestas aleatórias ao invés de árvore de decisão simples, bem como uma redução significativa no desvio padrão do R^2 nas três validações (3, 5 e 10 *folds)*.

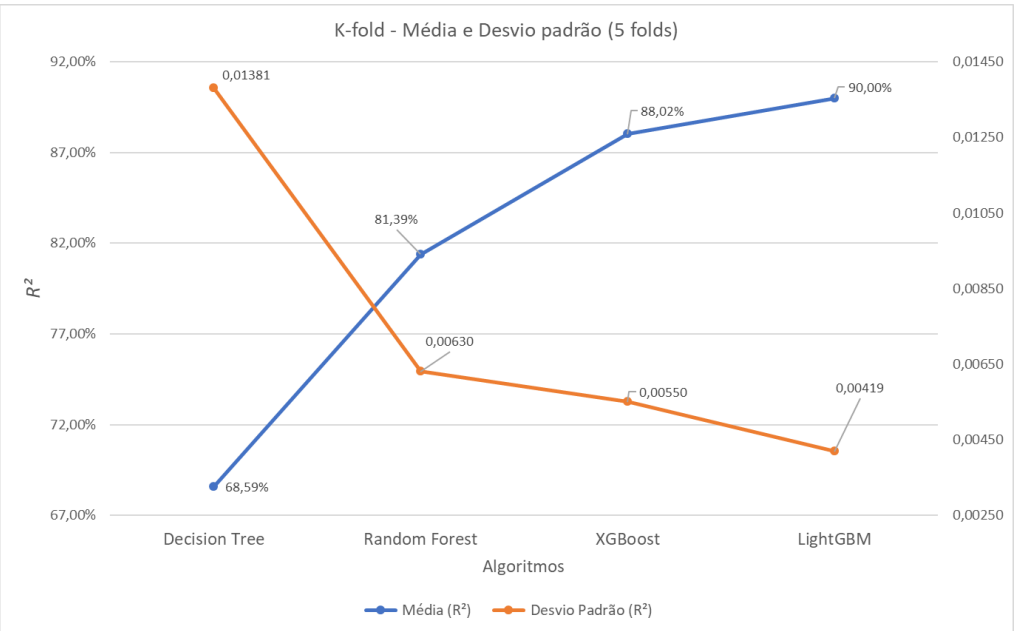
Os algoritmos *XGBoost* e *LightGBM* atingiram a meta de coeficiente de determinação definida neste projeto de 85%, apresentando ambos desvio padrão bastante menores que os outros dois algoritmos anteriormente citados em todos as três validações (3, 5 e 10 *folds*). Ambos os algoritmos tiveram resultados similares, tanto do ponto de vista de R^2 médio quanto o desvio padrão, destoando bastante dos dois primeiros algoritmos utilizados. Estas comparações se encontram ilustradas conforme as Figuras 5, 6 e 7.

Figura 5 – *K-fold*, 10 *folds*, média e desvio padrão (R^2) por algoritmo.



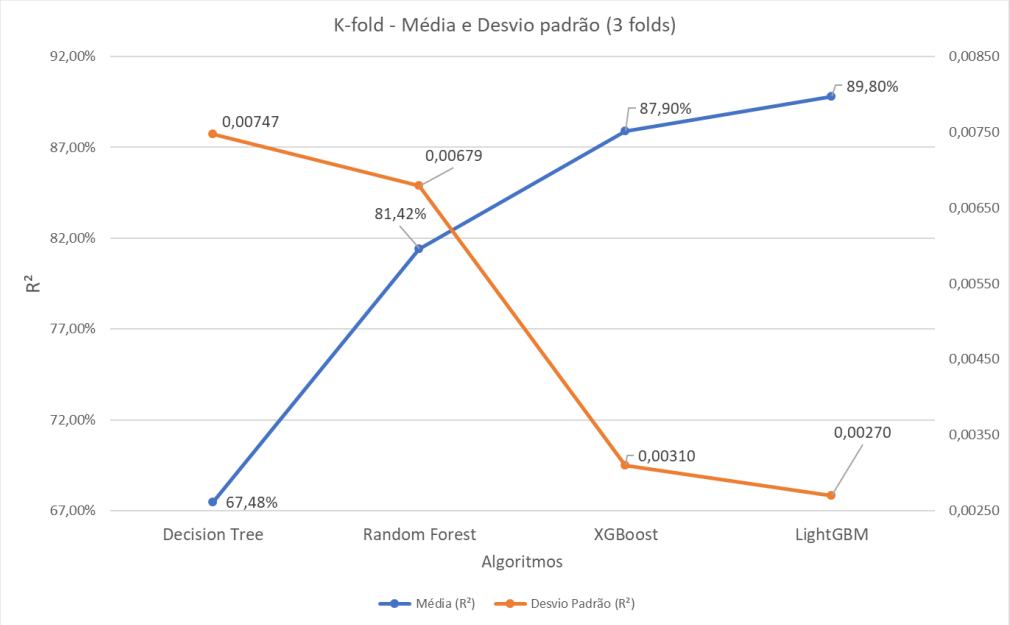
Fonte: Próprio autor (2022).

Figura 6 – *K-fold*, 5 *folds*, média e desvio padrão (R^2) por algoritmo.



Fonte: Próprio autor (2022).

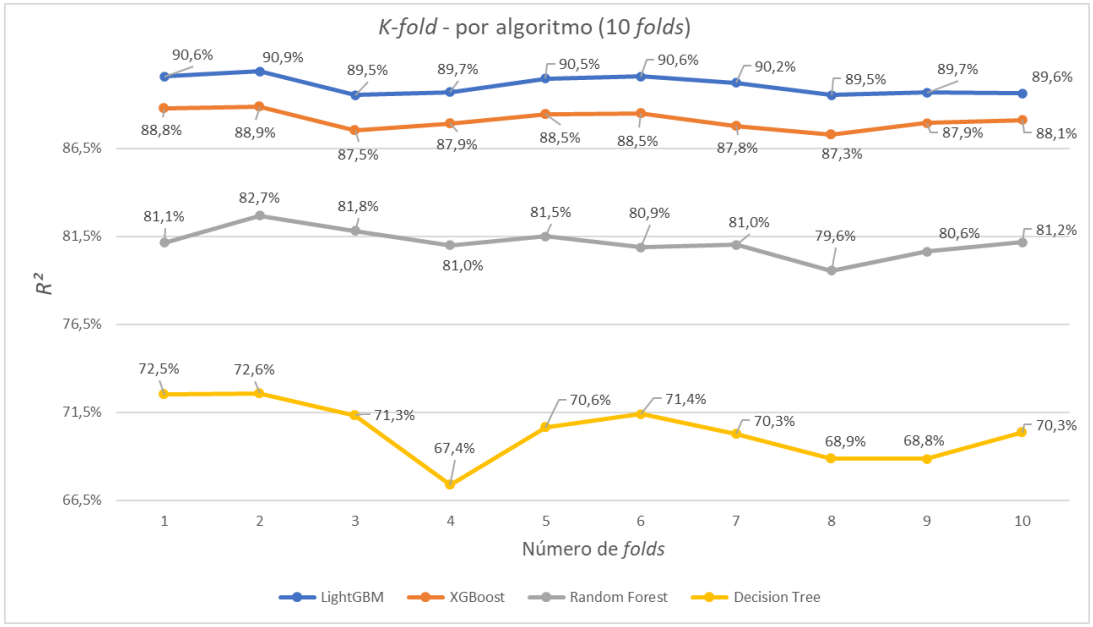
Figura 7 – *K-fold*, 3 *folds*, média e desvio padrão (R^2) por algoritmo.



Fonte: Próprio autor (2022).

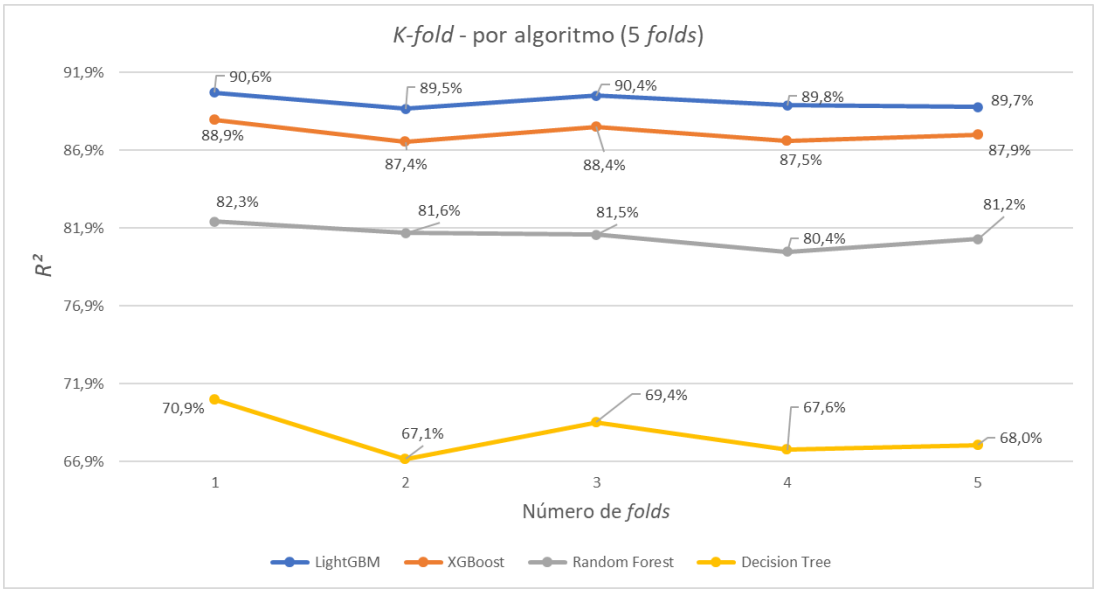
Analisando mais a fundo as três validações, é possível avaliar o R^2 de cada *fold* que compôs as médias e desvios padrões das Figuras 5, 6 e 7 anteriores. É possível verificar na Figura 8 que o algoritmo árvore de decisão simples obteve um *fold* destoante dos demais, *fold* de número 4. Isso se repetiu também na validação de 5 *folds*, quando se verifica o *fold* de número 2 na Figura 9, o que pode significar um período, no *dataset* de teste, onde as variáveis de entrada do modelo estavam bastante diferentes do período de aprendizado e a árvore de decisão simples não conseguiu predizer com robustez, diferente dos outros três algoritmos. Já na Figura 10, por ter menor número de folds, não foi possível identificar *fold* destoante dos demais em nenhum dos quatro algoritmos testados.

Figura 8 – *K-fold*, 10 *folds*, R^2 por *fold*.



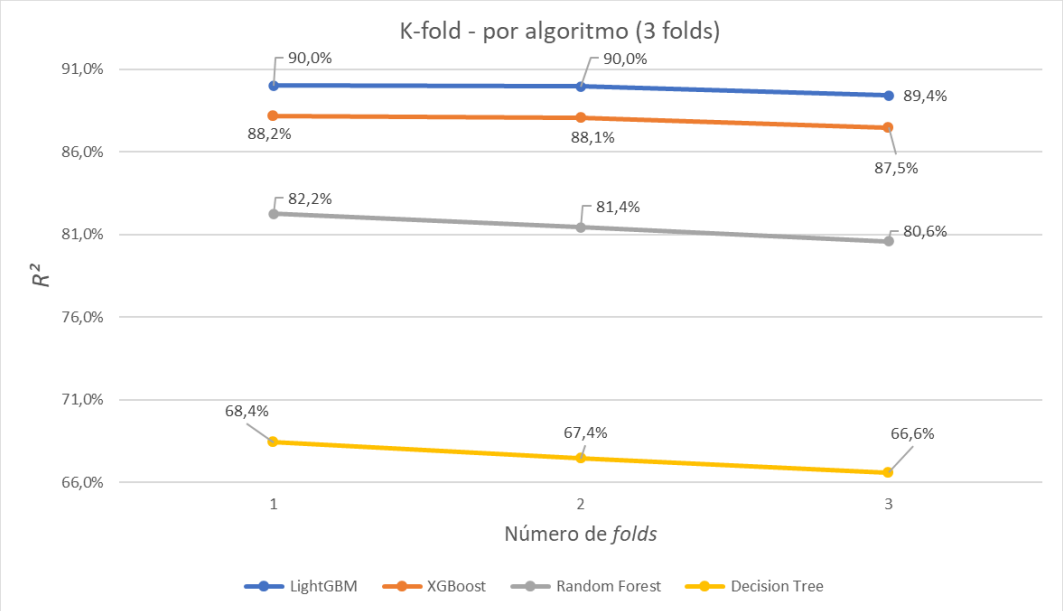
Fonte: Próprio autor (2022).

Figura 9 – *K-fold*, 5 *folds*, R^2 por *fold*.



Fonte: Próprio autor (2022).

Figura 10 – *K-fold*, 3 *folds*, R^2 por *fold*.



Fonte: Próprio autor (2022).

5.3 Resultados em dados de teste

Uma vez obtido resultados satisfatórios com a validação cruzada já realizada nos modelos com os dados de teste, se faz necessário a aplicação final dos algoritmos em dados de teste, tendo assim a validação definitiva necessária para implementação do melhor modelo, caso algum dos algoritmos atinja a meta estipulada de *RMSE* menor que 0,25% de oxigênio residual com um coeficiente de determinação superior a 85%.

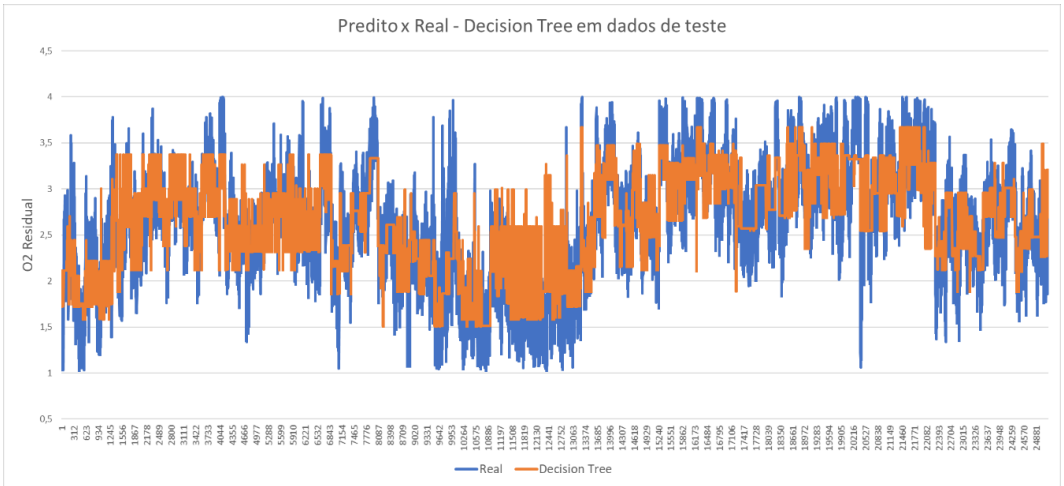
Os algoritmos foram submetidos as variáveis de entrada do *dataset* de teste (x_{teste}), realizando assim predições do valor de oxigênio residual ($y_{predito}$). Os resultados obtidos da comparação entre a saída predita de cada algoritmo e o robô analisador de oxigênio residual estão dispostos conforme a Tabela 9 e ilustrados individualmente segundo as Figuras 11, 12, 13 e 14.

Tabela 10 – Métricas de acurácia e R^2 em dados de teste.

Algoritmo	**Métricas de Acurácia**			***R²***
	MAE	***MSE***	***RMSE***	
Decision Tree	0,24665	0,10278	0,32060	0,73031
Random Forests	0,21055	0,07296	0,27011	0,80856
XGBoost	0,16554	0,04550	0,21332	0,88060
LightGBM	0,14910	0,03723	0,19296	0,90230

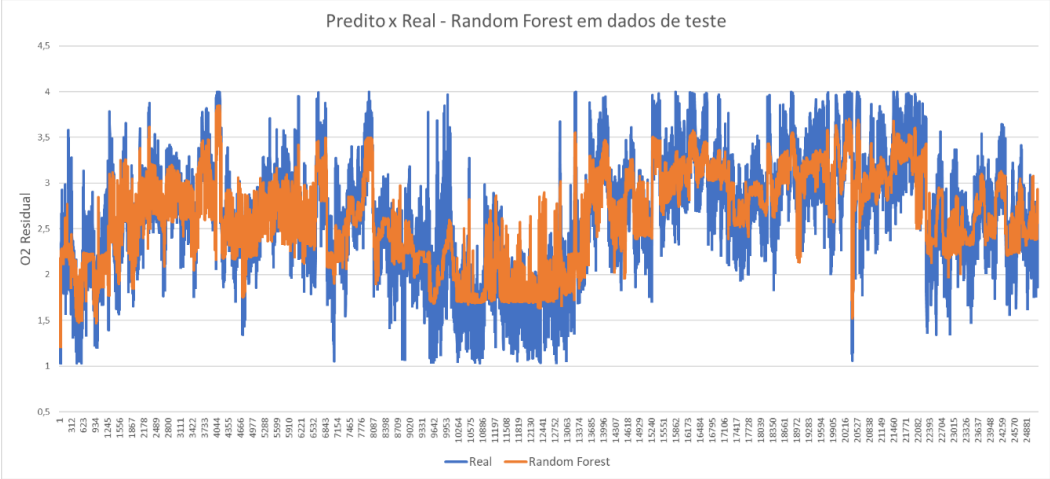
Fonte: Próprio autor (2022).

Figura 11 – Predito x sensor real – *Decision Tree*.



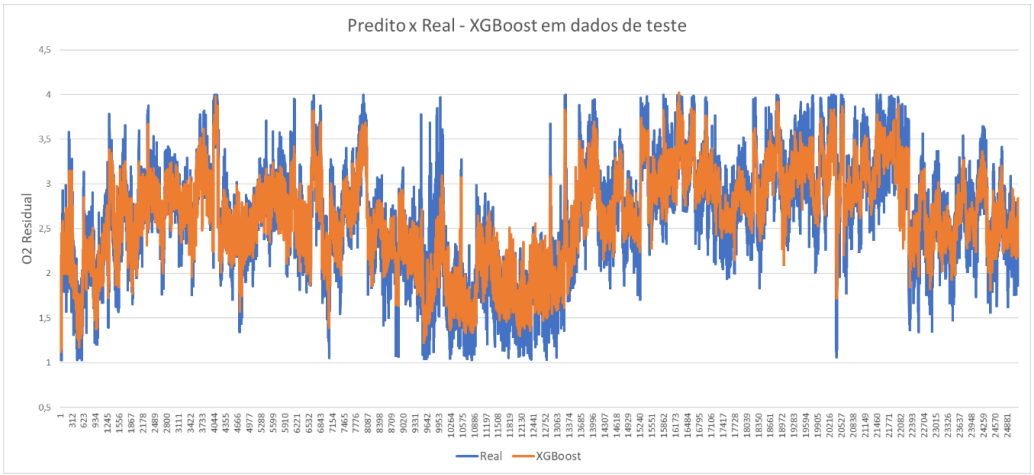
Fonte: Próprio autor (2022).

Figura 12 – Predito x sensor real – *Random forest*.



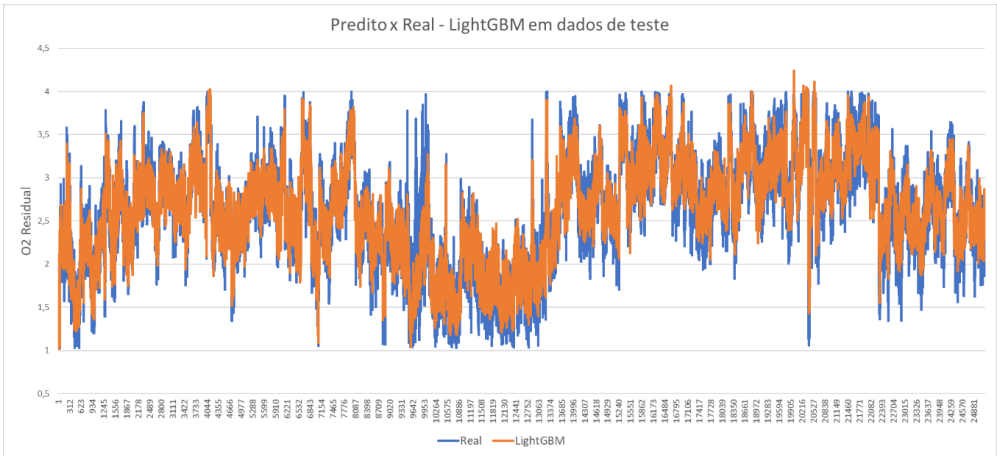
Fonte: Próprio autor (2022).

Figura 13 – Predito x sensor real – *XGBoost*.



Fonte: Próprio autor (2022).

Figura 14 – Predito x sensor real – *LightGBM*.



Fonte: Próprio autor (2022).

De posse de todos os resultados obtidos na construção de um sensor virtual de oxigênio residual utilizando aprendizado de máquina, se torna viável realizar a implementação prática deste sensor dentro do sistema de automação da fábrica em questão. Inicialmente é necessário realizar a exportação do algoritmo de melhor resultado, converter a linguagem *python* para a linguagem do sistema de automação (SDCD) da planta. Com o algoritmo implementado, inicialmente não conectado de nenhuma forma ao processo, é necessário realizar o treinamento dos operadores explicando o funcionamento do sensor e as entradas que são utilizadas pelo modelo.

Após um período de treinamento da equipe de operação, poderá ser utilizado o sensor como variável de processo em malhas de controles já existentes como também em estratégias novas de controle antecipativo com o objetivo de reduzir consumo de gás natural. Poderá este também servir de *cross check* com o sensor real e até mesmo como substituto caso o robô analisador venha a falhar e se torne indisponível, evitando assim uma parada da área de forno de cal e consequentemente um aumento do custo de produção.

6 CONCLUSÕES

A análise de oxigênio residual em forno de cal em fábricas de celulose é de grande importância para controlar o processo de combustão do forno e consequentemente controlar variável mais importante de entrega da área de forno de cal para o processo de produção de celulose, o carbonato de cálcio. Nesta etapa do processo a medição de oxigênio norteia quanto a relação combustível/comburente que, na fábrica em questão, é bastante complexa por realizar a utilização de três diferentes combustíveis.

O uso de combustível em excesso onera bastante o custo de produção por conta do preço de gás natural de petróleo, já a sua escassez pode reduzir muito a temperatura interna do forno, dificultando a conversão de carbonato de cálcio em óxido de cálcio e assim gerando problemas em outras áreas do processo. Por isso essa medição de oxigênio residual se faz tão fundamental para controlar as variáveis de processo do forno, sendo esta mensurada por um robô analisador que possui uma alta frequência de limpeza do sistema de amostragem por conta de intempéries do ambiente onde está submetido.

Com isso a medição de oxigênio residual é um grande candidato a ser virtualizado com a utilização de algoritmos baseado em aprendizado de máquina, aumentando com isso a robustez e a confiabilidade da medição, visto que é possível realizar comparações entre a medição real e a virtual. Também é possível utilizar o sensor virtualizado quando houver indisponibilidade do robô analisador, o que causaria indisponibilidade/inoperabilidade de toda a área, além de ser possível utilizar este sensor em estratégias preditivas de controle de processo, atuando antecipadamente (*feedforward*) sobre os combustíveis com o objetivo de redução do consumo destes, mantendo a carbonato de cálcio dentro do especificado pelo time de operação.

Desta forma com os resultados obtidos é possível responder às perguntas anteriormente feitas neste trabalho: "É possível utilizar algoritmos de AM, para a construção de um sensor virtual analisador de oxigênio residual que permitirá otimizar o processo de forno de cal?" e “Qual algoritmo de AM, dentre os escolhidos neste trabalho, se comportará melhor para predizer o volume de oxigênio residual de um forno?”. Foi apresentado que é possível realizar a criação de sensores virtuais analisador de oxigênio residual com quatro diferentes tipos de algoritmos, sendo o melhor desempenho dentre os testados aqui o *LightGBM*.

7 TRABALHOS FUTUROS

Pensando na continuidade deste trabalho, seria de grande valia realizar a implementação definitiva do melhor algoritmo (*LighGBM*) no sistema de automação (SDCD) da fábrica em questão, criando lógicas de comparação em tempo real entre o sensor virtual e o sensor real de oxigênio residual com a criação de um alarme por desvio alto. Este alarme poderia indicar um problema em alguma das medições de entrada do modelo, sinalizando a necessidade de uma calibração de algum instrumento, alertando a operação e consequentemente a manutenção do problema existente na área.

Outro ponto interessante é a implementação de uma lógica de chaveamento entre sensor virtual e sensor real caso haja indisponibilidade (congelamento de sinal, valor destoante etc.) do robô analisador. Isso pode mitigar ou até mesmo evitar indisponibilidade da área de forno de cal, fazendo com que o processo produtivo não pare ou não precise utilizar cal virgem de alto valor agregado. Outra grande vantagem seria a utilização deste sensor virtual como variável em estratégia de controle preditiva, manipulando o gás natural, único combustível que não é subproduto do processo, com o objetivo de reduzir o consumo ao máximo, mantendo obviamente a variável de carbonato de cálcio dentro das especificações de processo do time operacional.

Um passo futuro também seria a remodelagem dinâmica dos sensores baseados nos diferentes algoritmos utilizando uma janela móvel de dados, fazendo assim com que o as mudanças do físico-químicas do processo sejam constantemente aprendidas pelos sensores. Porém para este tipo de implementação se faz necessário a integração entre sistemas de automação (SDCD) e um sistema que tenha disponibilidade de linguagem *python* ou que o próprio sistema de automação já tenha *python* nativo, que não é o caso do sistema atual utilizado pela fábrica em questão.

REFERÊNCIAS

[1] Bajpai, P. (2015). **Basic overview of pulp and paper manufacturing process. In Green chemistry and sustainability in pulp and paper industry**. Publisher: Springer, 11-39.

[2] Theliander, H. (2009). **The Recovery of Cooking Chemicals: The White Liquor Preparation Plant.** Volume 2. Publisher: Gruyter, 335.

[3] Manning, R., & Tran, H. (2015). **Impact of cofiring biofuels and fossil fuels on lime kiln operation**. Publisher: Tappi Journal, 474-480.

[4] Xuefeng, Y. (2010). **Hybrid artificial neural network based on BP-PLSR and its application in development of soft sensors**. Publisher: Elsevier, 152-159.

[5] Anand, K., Mamatha, E., Reddy, C. S., & Prabha, M. (2019). **Design of neural network based expert system for automated lime kiln system**. Publisher: Journal Européen des Systèmes Automatisés, 369-376

[6] Juneja, P. K., & Ray, A. K. (2013). **Prediction based control of lime kiln process in a paper mill**. Publisher: Journal of Forest Products and Industries, 58-62.

[7] Gross, J., & Groß, J. (2003). **Linear regression**. Publisher: Springer, 33-38.

[8] Myles, A. J., Feudale, R. N., Liu, Y., Woody, N. A., & Brown, S. D. (2004). **An introduction to decision tree modeling**. Publisher: Journal of Chemometrics: A Journal of the Chemometrics Society, 18(6), 275-285.

[9] Breiman, L. (2001). **Random forests. Machine learning**. Publisher: Springer, 45(1), 5-32.

[10] Geurts, P., Ernst, D., & Wehenkel, L. (2006). **Extremely randomized trees. Machine learning**. Publisher: Springer, 63(1), 3-42.

[11] Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., & Cho, H. (2015). **Xgboost: extreme gradient boosting.** Publisher: JMLR: Workshop and Conference Proceedings, 1(4), 1-4.

[12] Friedman, J. H. (2001). **Greedy function approximation: a gradient boosting machine**. Annals of statistics. Publisher: Institute of Mathematical Statistics, 1189-1232.

[13] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). **Lightgbm: A highly efficient gradient boosting decision tree**. NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems, 30, 3146-3154.

[14] Lin, B., Recke, B., Knudsen, J. K., & Jørgensen, S. B. (2007). **A systematic approach for soft sensor development. Computers & chemical engineering**. Publisher: Elsevier, 31, 419-425.

[15] Pan, H., Su, T., Huang, X., & Wang, Z. (2021). **LSTM-based soft sensor design for oxygen content of flue gas in coal-fired power plant**. Publisher: Sage Journals, 43(1), 78-87.