UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Sensor virtual de alvura em polpa branqueada de celulose baseado em Inteligência Artificial

Kleverson Glauber Figueiredo de Paula

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 de alvura em polpa branqueada de celulose baseado em Inteligência Artificial

Kleverson Glauber Figueiredo de Paula

USP - São Carlos

2022

Kleverson Glauber Figueiredo de Paula

Sensor virtual de alvura em polpa branqueada 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. Renato Tinós

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)

P324s Paula, Kleverson Sensor virtual de alvura em polpa branqueada de celulose baseado em Inteligência Artificial / Kleverson Paula; orientador Renato Tinós. -- São Carlos, 2022.
55 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. Celulose. 2. Branqueamento. 3. Sensor virtual. 4. Alvura. 5. Inteligência Artificial. I. Tinós, Renato, 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

A minha esposa Dirciana e filho Glauber pela compreensão, carinho e apoio incansável.

AGRADECIMENTOS

Ao amigo e excelente profissional Khayan Marques Sobral. Sem seu apoio incondicional tanto no MBA quanto no trabalho do dia a dia, nada disso seria possível.

Ao Dr. Renato Tinós, professor orientador sempre atencioso, extremamente profissional, didático e cirúrgico em nos orientar da melhor forma.

Aos nossos gestores da ANDRITZ, Luis Binotto e Henrique Garcia, por investirem neste nosso sonho de se especializar na área de Inteligência Artificial e Big Data.

RESUMO

DE PAULA, K. G. F. **Sensor virtual de alvura em polpa branqueada 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.

Com a demanda crescente do consumo global de celulose branqueada, as indústrias deste setor têm sido cada vez mais demandadas a otimizarem seus processos para se manterem competitivas. A etapa de branqueamento é fundamental para alcançar as especificações exigidas pelos clientes. O atributo alvura da polpa de celulose é o principal entregável desta fase do processo, motivando assim a aplicação de inteligência artificial para predição deste tão importante atributo. Busca-se assim otimizar o consumo de químicos sem comprometer a qualidade do produto. Atualmente é utilizado um robô para efetuar a medição de alvura e outras características morfológicas da polpa de celulose. O resultado das análises é obtido a cada 30 minutos, permitindo assim que o controle clássico do tipo *feedback* efetue as devidas correções para manter a alvura medida no valor desejado. Neste trabalho, propõe-se utilizar Aprendizado de Máquina para predição da alvura da polpa de celulose. A escolha dos atributos relevantes para a criação do modelo de predição, bem como o período de histórico em estudo, foi realizada em conjunto com operadores e engenheiros de processo, sendo posteriormente validada com o uso de algoritmos computacionais estatísticos. Para construção dos modelos preditivos foram utilizados diferentes tipos de algoritmos baseados em árvores de decisão, avaliados e comparados a acurácia obtida e outros indicadores de desempenho entre eles. Foram comparados os algoritmos *Decision Tree*, *Random Forest*, *XGBoost* e *LightGBM*. Resultados experimentais indicam que algoritmos de Aprendizado de Máquina podem ser utilizados para gerar sensores virtuais de alvura eficientes.

Palavras-chave: celulose; branqueamento; sensor virtual; alvura; inteligência artificial; aprendizado de máquina.

ABSTRACT

DE PAULA, K. G. F. **Brightness virtual sensor based on Artificial Intelligence in pulp mill bleaching area**. 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.

With the growing demand from the global consumption of bleached pulp, industries in this sector have been increasingly required to optimize their processes in order to remain competitive. The bleaching stage is essential to reach the specifications required by customers. The Brightness attribute of cellulose pulp is the main deliverable of this stage of the process, thus motivating the application of artificial intelligence to predict this very important attribute. The aim is thus to optimize the consumption of chemicals without compromising product quality. Currently, a robot is used to measure the brightness and other morphological characteristics of the cellulose pulp. The analysis results are obtained every 30 minutes, thus allowing the classic feedback control to make the necessary corrections to keep the measured brightness at the desired value. In this work, will be used Machine Learning to predict the brightness of cellulose pulp. The choice of relevant attributes for creating the prediction model, as well as the historical period under study, was carried out in conjunction with operators and process engineers, and subsequently validated using statistical computational algorithms based on decision tree. To build the predictive models, different types of algorithms were used, evaluated and compared the obtained accuracy and other indicators between them. Decision tree, Random Forests, XGBoost and LightGBM algorithms were compared. Experimental results indicate that Machine Learning algorithms can be used to generate efficient whiteness virtual sensors.

Keywords: pulp; bleaching; virtual sensor; brightness; artificial intelligence; machine learning;

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE SÍMBOLOS

tsa/d	Tonelada seca ao ar por dia
kg/tsa	Quilograma por tonelada seca ao ar
t/h	Tonelada por hora
m^3/h	Metro cúbico por hora
ISO	Percentual de alvura em relação ao padrão
Kappa	Número obtido por medição indireta do percentual de lignina ainda restante na polpa
H2SO4	Ácido Sulfúrico
ClO2	Dióxido de cloro
pH	Potencial hidrogeniônico
°C	Grau Celsius

SUMÁRIO

1 INTRODUÇÃO

Atualmente, de forma global, as indústrias de celulose e papel têm sofrido mudanças decorrentes de alterações no cenário industrial com forte influência da rápida evolução de tecnologias. Os desafios deste setor vão desde a qualidade de matérias primas, passando por considerações ambientais e alterações em tecnologias do processo industrial. O desenvolvimento e o aperfeiçoamento das tecnologias representam uma forte possibilidade de garantir a competitividade das fábricas de celulose e papel.

O papel é produzido a partir de fibras celulósicas. A fabricação do papel consiste em criar uma superfície, a qual suas propriedades dependem das características das fibras utilizadas. Estas propriedades são definidas durante os processos de polpação e branqueamento [1]. Os fardos de celulose branqueadas de eucalipto são negociados como *commodity*, sendo assim, o preço destes depende fortemente de oscilações do mercado. Agregar valor, através da otimização da etapa de branqueamento, controlando características físico-químicas e estruturais das fibras, se torna uma alternativa para manter-se competitivo neste mercado.

1.1 Sensor virtual de alvura baseado em Inteligência Artificial

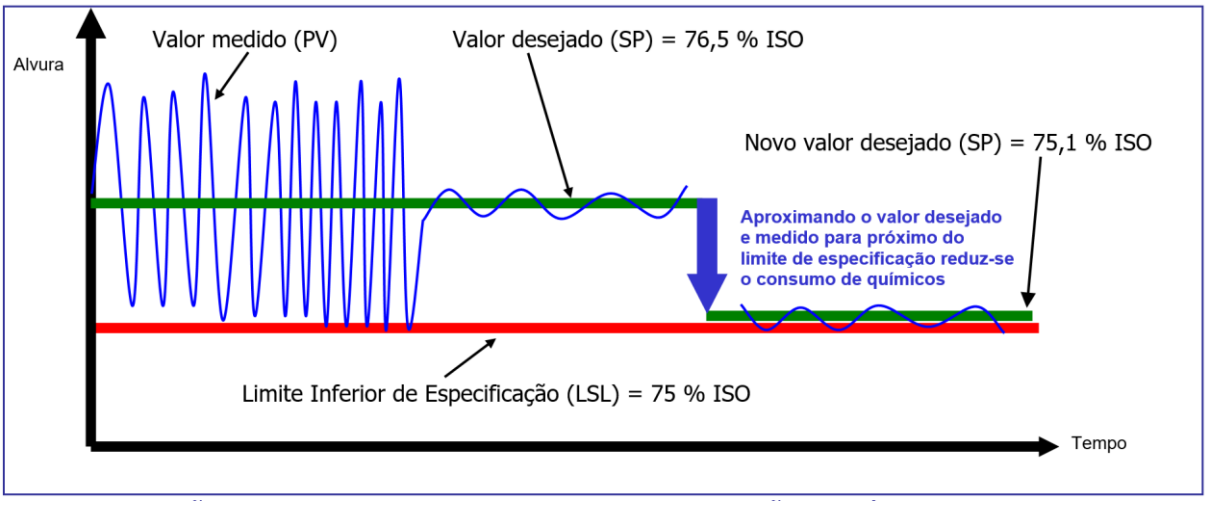
Vários problemas relacionados com o processo de produção da celulose são passíveis de serem otimizados e automatizados por meio do uso de Inteligência Artificial [2]. O problema que será tratado neste projeto é encontrado na etapa de branqueamento. Um dos principais motivadores para propor melhorias nessa etapa da fábrica é o fato dos químicos serem o maior custo de consumíveis do processo de fabricação, ficando atrás apenas da madeira e seguido do consumo de energéticos (combustíveis).

O desafio é controlar a variável alvura na saída do reator do primeiro estágio do branqueamento, buscando a redução de variabilidade, permitindo assim operar com o valor desejado o mais próximo possível do limite inferior de especificação, otimizando por consequência a adição de químicos conforme ilustrado na Figura 1. Este estágio denomina-se A/D0 e possui essa terminologia pelo fato de o mesmo utilizar os produtos químicos ácido sulfúrico e dióxido de cloro. O controle de alvura é desafiador, visto que a adição de químicos é feita na entrada do reator e apenas mede-se o resultado da dosagem aplicada, por meio da medição da alvura, após um tempo prolongado de retenção da polpa de celulose. A medição da

alvura é realizada por um robô que coleta amostras de 30 em 30 minutos na entrada e na saída do reator.

Para tornar o controle de alvura robusto, capaz de manter essa variável a mais próxima possível do limite inferior de especificação, a estratégia pensada foi desenvolver um modelo de predição baseado em Inteligência Artificial. Busca-se assim virtualizar o robô medidor de alvura, permitindo como consequência aumentar a frequência de aferimento da variável alvura e possibilitando ao controle realizar mais correções.

Figura 1 – Efeito da redução de variabilidade da alvura



Fonte: Próprio autor (2021).

Para o problema estudado, foram propostos e comparados neste trabalho vários tipos de algoritmos de aprendizagem de máquina (AM), que foram avaliados em relação à acurácia, poder de generalização, bem como outras métricas de avaliação. Validação cruzada foi aplicada na avaliação dos algoritmos.

1.2 Justificativa

O controle atual é feito por meio de um controlador clássico do tipo *feedback* que, devido às características já relatadas anteriormente, pode atuar tardiamente para correção da alvura. Como consequência, pode-se ocorrer o excesso ou falta de químicos, além do risco de desclassificação do produto final em função do mesmo não atender às especificações de alvura do cliente.

A proposta desse projeto é investigar o desenvolvimento de um sensor virtual de alvura utilizando algoritmos de Aprendizado de Máquina (AM) supervisionados. Espera-se que o sensor virtual permita predizer a alvura na saída do reator a cada 1 minuto, possibilitando ao controlador a correção da dosagem de químicos no processo de forma antecipada (*feedforward*) e, portanto, mais otimizada. Ficará a cargo do atual controle do tipo *feedback* fazer pequenas correções remanescentes de eventuais erros do modelo de predição.

Outro benefício esperado com o sensor virtual é que ele sirva de *cross check* do sensor real (robô), permitindo encontrar possíveis defeitos embrionários e informando inconsistências nas variáveis de entrada que o modelo utiliza, no sensor real ou até mesmo indicando um erro do modelo utilizado. Essa estratégia é muito interessante porque permite identificar defeitos no sistema antes que ele se agrave ao ponto de gerar uma falha e assim provocar perdas maiores no processo como consumo elevado de químicos ou até mesmo um produto fora da especificação (desclassificado) na etapa final do processo.

Um benefício adicional é utilizar apenas o sensor virtual em caso de falha do sensor real (robô) por tempo prolongado. Tal uso é especialmente interessante pois a reposição de peças do robô é bastante cara e lenta devido ao fato de serem importadas da Finlândia. Dessa forma não será necessário desligar o controlador durante o período de manutenção, o que provocaria alto consumo de químicos. Além disso, o uso do sensor virtual proporcionaria também evitar o armazenamento de peças do robô de custo elevado em estoque.

A aplicação de algoritmos de AM poderá permitir um processo de controle de alvura mais robusto, eficiente, entregando para o cliente final uma alvura a mais próxima possível do limite inferior de especificação, e consequentemente tendo um menor gasto com produtos químicos e com manutenção dos robôs.

1.3 OBJETIVOS

A Hipótese investigada neste trabalho é que “com um conjunto de treinamento suficientemente grande para o processo de branqueamento, o algoritmo de AM supervisionado conseguirá predizer a alvura da polpa de celulose na saída do estágio A/D0 do branqueamento”. Espera-se que o erro de predição para os conjuntos de teste seja suficientemente baixo para conferir robustez ao processo, permitindo desenvolver um sensor virtual capaz de ser utilizado em uma estratégia antecipativa (*feedforward*) somada à estratégia *feedback* do sensor real (robô), reduzindo assim a variabilidade do controle de alvura.

O objetivo principal do trabalho é, portanto, desenvolver um sensor virtual inteligente para a alvura da polpa de celulose na saída do estágio A/D0 de branqueamento. Espera-se que o sensor virtual inteligente permita trabalhar bem mais próximo do limite inferior de especificação e consequentemente, obter-se a minimização do consumo de produtos químicos.

1.3.1 Específicos

A principal questão de pesquisa que será tratada neste projeto é: "É possível utilizar algoritmos de AM, como descritos anteriormente, para a construção de um sensor virtual de alvura eficiente que permitirá otimizar o processo de branqueamento?". Outra questão que será tratada é: “Qual algoritmo de AM representará o melhor desempenho para o problema proposto?”

Para responder a essas perguntas neste trabalho são utilizados indicadores de desempenho para avaliar a acurácia e o poder de generalização dos modelos obtidos pelos diferentes algoritmos. Visando obter um alto desempenho na estratégia de controle, o principal indicador de desempenho, bem como a meta de acurácia estipulada para o modelo é o *RMSE* (*root mean squared error*) menor que 0,3%.

Para avaliar o poder de generalização do modelo é utilizado o indicador de desempenho de coeficiente de determinação, R^2 no processo de validação cruzada obtido através do algoritmo *K-fold*, com uma meta superior a 80%.

2 FUNDAMENTAÇÃO TEÓRICA

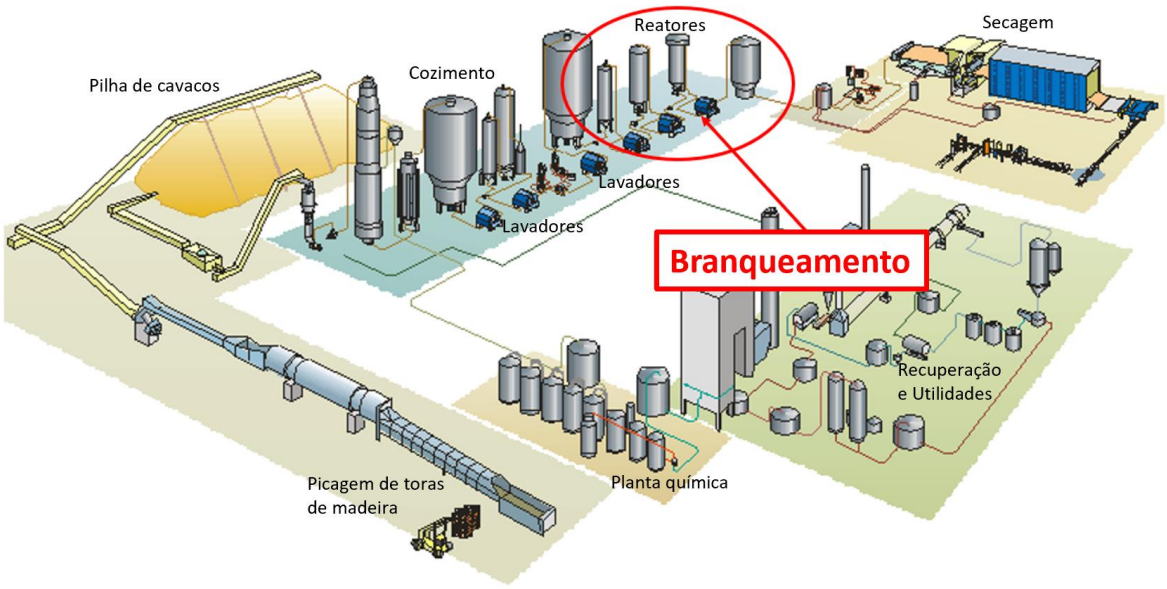
O processo produtivo de celulose branqueada é divido em duas grandes ilhas, sendo a primeira a parte florestal e a segunda a extração da celulose (industrial). A área florestal tem como objetivo o fornecimento de madeira apropriada para a extração de celulose. Essa ilha abrange desde o processo de clones das mudas de eucalipto em viveiros, passando pelo plantio e finalizando na colheita e descascamento das toras de madeira após cerca de 6 a 8 anos de cultivo.

2.1 Processo *Kraft* de celulose branqueada

A área industrial tem início na fase de preparação de cavacos, sendo este produzido por meio da trituração das toras de madeira que são a matéria prima oriundas das florestas de eucalipto. Este microprocesso é puramente mecânico, onde ocorre a picagem das toras e o armazenamento. Os cavacos provenientes da picagem possuem dimensões especificadas e são armazenados na pilha, a qual alimenta o digestor [3]. Os cavacos fora de especificação servem de combustível e são direcionados para a pilha de alimentação da caldeira de biomassa, que possui a função de gerar vapor de alta pressão (~92 bar) o qual, posteriormente, será utilizado em um turbo gerador para geração de energia elétrica.

Os cavacos serão utilizados no digestor, com a finalidade de promover o cozimento dessa madeira, utilizando produtos químicos altamente alcalinos em conjunto com vapor de média (~12 bar) e baixa pressão (~4 bar), obtendo após um tempo de retenção a polpa marrom de celulose. Em seguida essa polpa é lavada, deslignificada e depurada, sendo entregue à etapa de foco deste trabalho, o branqueamento. Uma vez tendo sido branqueada a polpa de celulose, se faz necessário promover a remoção de água para viabilizar o transporte até os clientes. Essa etapa é denominada secagem, possuindo microprocessos como formação de folha, prensagem, secagem a vapor de baixa pressão, formação de fardos e embalagem final. Está ilustrado na Figura 2 uma planta de produção de celulose com as etapas do processo.

Figura 2 – Visão geral de um processo produtivo de celulose branqueada.



Fonte: Modificado de ANDRITZ – GMBH (2007).

2.2 Processo de branqueamento da polpa

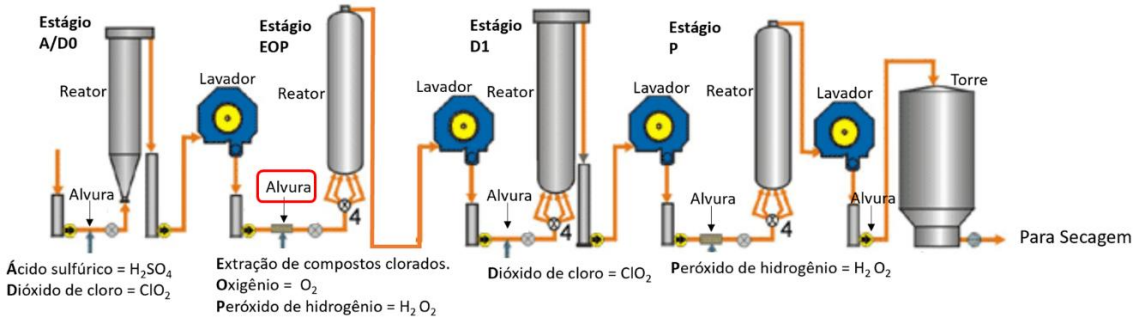
O objetivo do branqueamento da polpa é dar continuidade à deslignificação iniciada no digestor e, por meio de oxidantes, remover qualquer lignina residual que permaneça após a etapa de cozimento. Essa lignina residual não é removida durante o processo de cozimento, se assim fosse, comprometeria o rendimento e as propriedades físico-químicas das fibras [4].

A etapa do branqueamento tem por finalidade também melhorar o brilho e a limpeza da celulose para atender às exigências dos clientes. As variáveis alvura e sujidade são as mais relevantes para o atendimento das especificações de venda. A alvura da polpa é medida como a capacidade de uma folha de celulose refletir a luz direcionada a ela e isso é afetado tanto pela absorção quanto pela dispersão da luz na folha. O comprimento de onda utilizado para essa medição é de 457 nm, comparado à medida padrão de refletância do óxido de magnésio, gerando um percentual de alvura ISO [5].

O processo de branqueamento é comumente dividido em estágios, primeiramente para promover a remoção da lignina residual ainda presente na polpa de celulose, e posteriormente promover o seu alvejamento gradual, minimizando as perdas decorrentes da degradação das fibras proveniente do ataque químico provocado pelos alvejantes. A etapa de branqueamento

da fábrica em estudo possui 4 estágios sequenciais tratando a polpa nos reatores, fazendo o uso de produtos químicos e lavando em seguida nos lavadores para remoção de compostos químicos remanescentes da reação. Os estágios do branqueamento são: A/D0, EOP, D1 e P, sendo as abreviaturas em função dos produtos químicos utilizados conforme ilustrado na Figura 3, onde é possível observar também a medição da alvura efetuada de forma automatizada por robôs em cada um dos estágios. A alvura destacada em vermelho refere-se à alvura a ser predita.

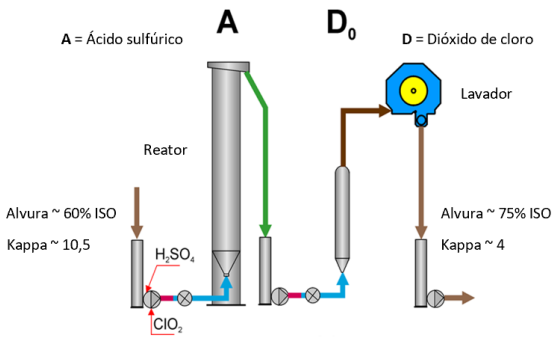
Figura 3 – Visão geral do branqueamento.



Fonte: Modificado de ANDRITZ – GMBH (2004).

O estágio A/D0 tem a finalidade de remover o residual de lignina, utilizando para isso os produtos químicos dióxido de cloro e ácido sulfúrico, bem como adição de vapor de média pressão e tempo de residência em reator. Essa remoção da lignina é indiretamente medida através do número Kappa obtido por um robô. Após cada estágio ocorre a lavagem da polpa de celulose para remoção dos químicos residuais e minimização do consumo elevado de químicos da etapa seguinte, por conta do pH que em cada estágio alterna entre ácido e alcalino. O ganho de alvura obtido nessa etapa é da ordem de 15% ISO, deixando o estágio com uma alvura de polpa em torno de 75% ISO, como mostrado na Figura 4.

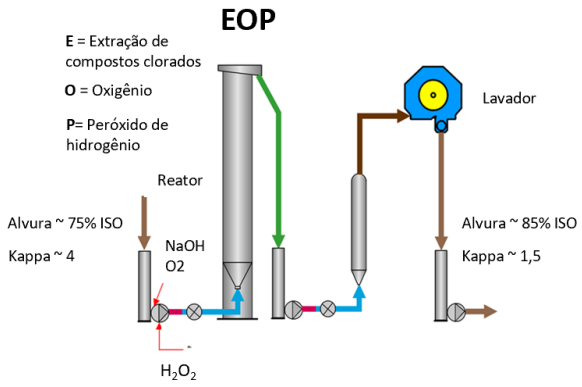
Figura 4 – Estágio A/D0 do branqueamento.



Fonte: Modificado de ANDRITZ – GMBH (2007).

Já no estágio EOP, são utilizados os químicos soda cáustica, oxigênio e peróxido de hidrogênio, com o objetivo de extrair os compostos clorados do estágio anterior e elevar a alvura da polpa para em torno de 85% ISO, como mostrado na Figura 5.

Figura 5 – Estágio EOP do branqueamento.



Fonte: Modificado de ANDRITZ – GMBH (2007).

Nos estágios seguintes a alternância de produtos químicos ocorre novamente, com dióxido de cloro e ácido sulfúrico no estágio D1, elevando-se a alvura ao patamar próximo de 88% ISO. No último estágio, denominado P, utiliza-se o peróxido de hidrogênio juntamente com a soda cáustica, tendo como principal função estabilizar a alvura, prevenindo a reversão precoce desta no produto final e elevando-a para em torno de 90% ISO.

Todos os estágios possuem reatores químicos com a função de promover o tempo de residência para que ocorram as reações químicas completas necessárias para o processo de deslignificação e alvejamento da polpa de celulose. Este tempo varia com a produção do branqueamento, oscilando entre 2h e 5h.

2.3 Regressão e principais algoritmos de aprendizado de máquina baseados em árvore de decisão

Em uma investigação científica se formulam hipóteses sobre relacionamento entre variáveis independentes ou de entrada (**x**) que possam explicar totalmente ou parcialmente uma variável dependente ou variável de saída (y), podendo assim formar um modelo de regressão a fim de realizar predições de y com base em valores conhecidos de **x**. Este modelo pode ser considerado linear, quando as variáveis possuem um relacionamento linear ou

aproximadamente linear entre elas, ou pode ser considerado não linear quando as variáveis não possuem um relacionamento linear [6]. O modelo linear de regressão pode ser aplicado quando todas as variáveis de entrada forem quantitativas.

São utilizadas algumas medidas de acurácia para avaliação do modelo, tais como o coeficiente de determinação (R^2), erro médio absoluto (*MAE*), erro médio quadrático (*MSE*) e raiz quadrada do erro médio quadrático (*RMSE*), conforme as equações a seguir. Essas medidas pontuam a disparidade entre a saída predita e a saída real sendo o *RMSE* a principal medida escolhida como alvo para este trabalho. O *RMSE* é interessante porque penaliza mais severamente valores discrepantes em relação à média.

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

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

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

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

Sendo que *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.

As árvores de decisão (*Decision Tree*) são métodos de aprendizado de máquina para inferência de regras de decisão simples que recebe um vetor com *n* valores (*features*) como entrada e retorna uma decisão (saída). É um algoritmo de aprendizado de máquina supervisionado utilizado para classificação ou regressão sendo uma das abordagens de modelagem preditiva mais usadas em estatística, mineração de dados e aprendizado de máquina [7]. Esse algoritmo, quando aplicado em problemas de regressão, tem como estrutura folhas, que representam rótulos de classe, e ramos que representam conjunções de características que levam a esses rótulos de classe [7].

As *Decision Trees*, as quais a variável dependente pode assumir valores contínuos, são chamadas de árvores de regressão e estão entre os algoritmos de aprendizado de máquina mais

populares devido à sua inteligibilidade, simplicidade e facilidade de implementação. Porém, como desvantagem, tem uma forte propensão ao sobre ajuste (*overfitting*) [8].

Os algoritmos de construção dessas árvores costumam funcionar de cima para baixo, selecionando em cada etapa uma variável que melhor divide o conjunto de itens. Diferentes algoritmos utilizam métricas distintas para mensurar o melhor modelo, medindo a homogeneidade da variável de saída dentro dos subconjuntos. Alguns exemplos de métodos utilizados na construção das árvores são a impureza de Gini e o ganho de informação. Essas são aplicadas em cada subconjunto de dados e os resultados são combinados para fornecer uma medida da qualidade da divisão [8].

Assim como a *Decision Tree*, o algoritmo de *Random Forest* é um algoritmo de aprendizado de máquina supervisionado utilizado para classificação ou regressão que possui a característica de combinar a simplicidade das *Decision Trees* com a flexibilidade e aleatoriedade para melhorar a precisão. Isso ocorre, pois as *Decision Trees* são propensas a sobre ajustarem (*overfitting*), na medida em que o tamanho da árvore/complexidade aumenta com os dados de treinamento, podendo prejudicar assim o poder de generalização do modelo [9].

Na *Random Forest*, múltiplas árvores são criadas pelo algoritmo utilizando-se de métodos de junção (*ensemble*) dos diferentes atributos e conjunto de dados contidos na base de dados de forma aleatória, ao invés da seleção partir do cálculo de impureza usado como critério em uma *Decision Tree* simples. Essa estratégia tende a reduzir os efeitos do sobre ajuste.

O número de árvores é definido através de um hiperparâmetro chamado número de estimadores, utilizando o resultado de cada modelo na definição de um único resultado, obtendo assim um valor final único (ex. por média). Em comparação com a *Decision Tree* simples, a *Random Forest* exige maior poder computacional, diretamente proporcional ao número de estimadores definido. Entretanto uma das maiores vantagens dessa floresta é o aumento da robustez quanto ao sobre ajuste [10].

O nome *XGBoost* vem de *eXtreme Gradient Boosting*, e representa uma categoria de algoritmos baseada em árvores de decisão com aumento de gradiente (*Gradient Boosting*). Aumento de gradiente significa que o algoritmo usa o artifício de *Gradient Descent* para minimização da perda (*loss*) propiciando que novos modelos sejam adicionados. O princípio do *Gradient Boosting* é a capacidade de combinar resultados de muitos classificadores de vieses fracos, tipicamente *Decision Trees*, que se combinam para formar algo parecido com um “forte comitê de decisão” [11].

Este algoritmo é altamente adaptável, uma vez que possui um grande número de hiperparâmetros passíveis de sintonia. É possível ajustar adequadamente o *XGBoost* para os mais variados tipos de problemas. Este algoritmo tem como funcionamento a criação de árvores de forma sequencial, tomando como base os resíduos anteriores (*boosting*) e posteriormente realizando a derivada (*Gradient*) do erro atual do conjunto (*ensemble*) [12].

O algoritmo *LightGBM* é um algoritmo também baseado em árvore de decisão que é projetado para ter as seguintes vantagens sobre o *XGBoost*: promover maior velocidade de treinamento, maior eficiência, menor uso de memória, melhor precisão, suporte de aprendizagem paralela, usar *GPU* e ter maior capacidade para lidar com dados em larga escala de múltiplas bases de dados simultâneas.

Existem duas estratégias diferentes para computar as árvores: *level-wise* utilizada pelos algoritmos *XGBoost* e *Random Forest* e *leaf-wise* utilizada para compor o *LightGBM*. A estratégia *level-wise* aumenta a árvore nível a nível, cada nó divide os dados priorizando os nós mais próximos da raiz da árvore. A estratégia *leaf-wise* faz crescer a árvore dividindo os dados nos nós com a maior mudança de perda. O crescimento em *level-wise* geralmente é melhor para conjuntos de dados menores, ao passo que o *leaf-wise* tende a ajustar-se em excesso podendo levar ao sobre ajuste. O crescimento em *leaf-wise* tende a se destacar em conjuntos de dados maiores, onde é consideravelmente mais rápido do que o crescimento em *level-wise*, isso se dá por conta deste algoritmo ser mais otimizado em relação ao *XGBoost* [13].

Outra característica do *LightGBM* é que além de utilizar a estratégia *leaf-wise*, ele usa uma abordagem diferente, a aproximação da divisão por meio da construção de histogramas dos atributos, dessa forma, o algoritmo não precisa avaliar cada valor único dos atributos para calcular a divisão, mas apenas os *bins* do histograma, que são limitados. Essa abordagem é muito mais eficiente para grandes conjuntos de dados, sem afetar adversamente a precisão.

3 TRABALHOS RELACIONADOS

Existem muitos trabalhos que tem como objetivo implementar aprendizado de máquina e inteligência artificial para a construção de sensores virtuais, nos mais diversos setores industriais do mundo. Aqui, o objetivo não é fazer uma revisão de literatura vasta sobre o tema, mas sim discutir trabalhos que visam a construção de sensores virtuais por meio de inteligência artificial em problemas semelhantes ao investigado neste trabalho. Para isso, foram selecionados dois trabalhos principais que se correlacionam com a virtualização de robô analisador em indústria de celulose e papel.

Gomes [14] utilizou metodologias estatísticas para validação de analisadores contínuos na indústria de celulose, além da utilização em *python* de regressão linear, *Decision tree* e *Random Forest* para a construção de um modelo a fim de predizer variáveis medidas por analisadores contínuos, objetivando robustez ao processo com o aumento da confiabilidade das medições. O melhor desempenho encontrado foi R^2 igual a 0,623. O autor cita que melhores resultados poderiam ser obtidos se não fosse a quantidade baixa de dados disponíveis no conjunto de dados (*underfitting*).

No trabalho aqui proposto são utilizadas estratégias de predição com o uso de algoritmos de inteligência artificial para o problema de predição e alvura, sendo algumas estratégias semelhantes às empregadas em Gomes (2021). É esperado, porém que não haja sérios problemas (*underfitting*) relacionados à baixa quantidade de dados disponíveis no conjunto de dados, uma vez que o histórico de dados obtidos para a realização do trabalho é de um período superior a 8 anos, sendo cada atributo (14 no total) medido em um intervalo médio de 30 minutos, totalizando mais de 90.000 instâncias (linhas de dados), sugerindo uma quantidade satisfatória de dados para a modelagem. A preocupação neste caso é com o sobre ajuste (*overfitting*), onde a estratégia para mitigá-lo é tratar os outliers dos atributos, bem como sintonizar os hiperparâmetros dos algoritmos adequadamente.

Domingues [15] comparou sensores virtuais analisadores de resistência à tração do papel, construídos com a utilização de regressão linear multivariada, regressão não linear por mínimos quadrados parciais (*PLS*) e redes neurais artificiais (RNA). Os modelos desenvolvidos apresentaram um baixo desempenho na previsão da variável de interesse quando foram empregados o algoritmo de regressão linear multivariada (R^2 igual a 0,358) e regressão não linear por mínimos quadrados parciais (R^2 igual a 0,358). Foram utilizadas técnicas de redução de dimensionalidade como análise da componente principal (*PCA*) e ainda assim o resultado

obtido foi insatisfatório. Após a utilização das redes neurais artificiais, foi possível atingir um R^2 de 0,89 mostrando ser este o modelo de melhor poder de generalização dentre os utilizados.

No trabalho aqui proposto, espera-se obter bons resultados com os modelos de regressão baseados em inteligência artificial no comparativo com os modelos estatísticos, tendo em vista que serão utilizados algoritmos que possuem maior capacidade de combinar resultados dentre vários (*ensemble*), minimizando os erros decorrentes dos modelos individuais. Por este motivo optou-se pelo uso dos algoritmos *Random Forest*, *XGBoost* e *LightGBM*. Outro fator determinante citado por Domingues [15] é que um dos motivos prováveis que culminaram no baixo desempenho dos algoritmos de regressão multivariada e regressão não linear por mínimos quadrados parciais possa ter sido a ausência de um dos principais atributos no conjunto de dados para a predição de resistência à tração do papel que é a densidade aparente do cavaco. No trabalho aqui proposto espera-se que a densidade aparente do cavaco não seja tão relevante para o algoritmo de predição de alvura nos estágios da etapa de branqueamento.

4 MATERIAIS E MÉTODOS

Os dados foram obtidos com o consentimento dos gestores da fábrica Veracel Celulose SA, localizada na cidade de Eunápolis no extremo sul da Bahia. Esta é uma fábrica que iniciou sua produção em 2005, em linha única com quatro estágios de branqueamento, com capacidade produtiva de cerca de 1,1 milhão de toneladas de celulose seca ao ar por ano.

Para desenvolver um sensor virtual de um robô analisador de alvura com o objetivo de utilizá-lo em uma malha de controle, otimizando dessa forma o consumo de químicos utilizados na etapa do processo de fabricação de celulose chamada branqueamento, propõe-se técnicas de AM. A primeira fase de construção do algoritmo consiste no pré-processamento que abrange as etapas de extração, transformação e carga dos dados.

Na etapa de extração dos dados, o objetivo principal é preparar os dados de forma a serem utilizados na etapa seguinte de extração de padrões. Para isso foi necessário extrair as informações de um banco de dados temporal (*datalake*) da fábrica em questão, disponíveis no sistema de gestão de informações da planta (*PIMS - Plant Information Management Systems*). O tempo de varredura da coleta das informações da fábrica é de, em média, 5 segundos, podendo variar de acordo com o tempo de resposta de cada atributo.

Por se tratar de um robô analisador de alvura que realiza uma medição e análise a cada 30 minutos, o tempo de varredura escolhido para extração dos dados e construção dos *datasets* foi definido como agregação, média de 30 minutos. O período para extração dos dados foi de 8 anos, de janeiro de 2013 a novembro de 2021, após consenso com a equipe de engenheiros químicos e florestais, por conta do dinamismo do processo bioquímico, fruto de constantes alterações no principal insumo utilizado no processo (madeira) em função de variações climáticas, tipos de solos diferentes por regiões de plantio e tipos de clones das mudas de eucalipto. Esse período prolongado de 8 anos de histórico de dados deve ser suficiente para se conseguir uma modelagem que abranja grande parte dos cenários possíveis, contribuindo assim com o poder de generalização e acurácia dos algoritmos.

A escolha de quais atributos utilizar para realização das modelagens foi feita por meio de entrevistas com os operadores dos cinco turnos que trabalham vinte e quatro horas por dia, sete dias por semana e com os engenheiros de processos, que possuem o conhecimento operacional de quais variáveis impactam naquela a ser virtualizada, que neste caso é a alvura na saída do primeiro estágio de branqueamento.

Foi preciso deslocar duas horas no tempo o atributo a ser virtualizado deslocando as instâncias desta variável em 4 (2h), tendo em vista que este se encontra fisicamente na saída do reator químico afetado por um tempo de retenção utilizado para maximizar a eficácia da reação química, calculado em função da vazão de entrada e do volume do reator. Desta forma ao fazer este deslocamento temporal, todas as variáveis se encontraram na mesma base de tempo. Foram mapeados como importantes quatorze atributos de entrada e um atributo de saída (alvura), conforme mostrado na Tabela 1.

Tabela 1 – Atributos do *dataset*.

Tag	**Descrição**	**Unidade**	**Mínimo**	**Máximo**
3205FI003.PNT	Produção do branqueamento	tsa/d	0	4000
3205FC030.MEAS	Vazão de vapor	t/h	0	30
3205LI020.PNT	Nível torre de massa	%	0	100
3205NC008.MEAS	Consistência da massa	%	0	15
3205FC027.MEAS	Vazão de massa	m³/h	0	1200
3205QI007.PNT	Kappa de entrada	-	0	15
3205FC032.MEAS	Vazão H2SO4	m³/h	0	25
3205FC032LOGA.RO01	Carga de H2SO4	kg/tsa	0	20
3205FC034.MEAS	Vazão de ClO2	m³/h	0	200
3205FAT-KAPPA-APC-DO.PNT	Fator Kappa	-	0	1
3205FC034LOGA.RO01	Carga de ClO2	kg/tsa	0	40
3205TC028.MEAS	Temperatura da massa	°C	0	100
3205QC037.MEAS	pH da massa	-	0	14
3205ALVURA5.RO01	Alvura de entrada do reator	%	0	100
3205QI044A.PNT	Alvura de saída do reator	%	0	100

Fonte: Próprio autor (2022).

Após a exportação dos dados do *PIMS*, se fez necessário o tratamento dos dados através do uso de técnicas de filtragem, visualizado os dados de cada atributo através dos gráficos *boxplot*, foram percebidos os *outliers* e as regiões do gráfico denominadas como primeiro e terceiro quartil (Q1 e Q3), tendo entre eles a faixa interquartil (FIQ). Utilizou-se o recurso de mapa de calor para validar a correlação entre os atributos e a variável a ser predita, confirmando assim a escolha dos atributos baseado na experiência dos operadores e engenheiros de processo.

Foram removidas as instâncias de cada atributo que continham *outliers* segundo as equações (05), (06) e (07) conforme Cavanha (1996). Após a remoção dos *outliers*, executou-se novamente a visualização dos dados através dos gráficos *boxplot* para cada atributo buscando-se analisar o resultado da exclusão. Instâncias contendo dados faltantes, dados inconsistentes e dados iguais oriundos de congelamento de sinal devido a falha de comunicação entre *datalake* e sistemas de automação foram removidas para cada atributo.

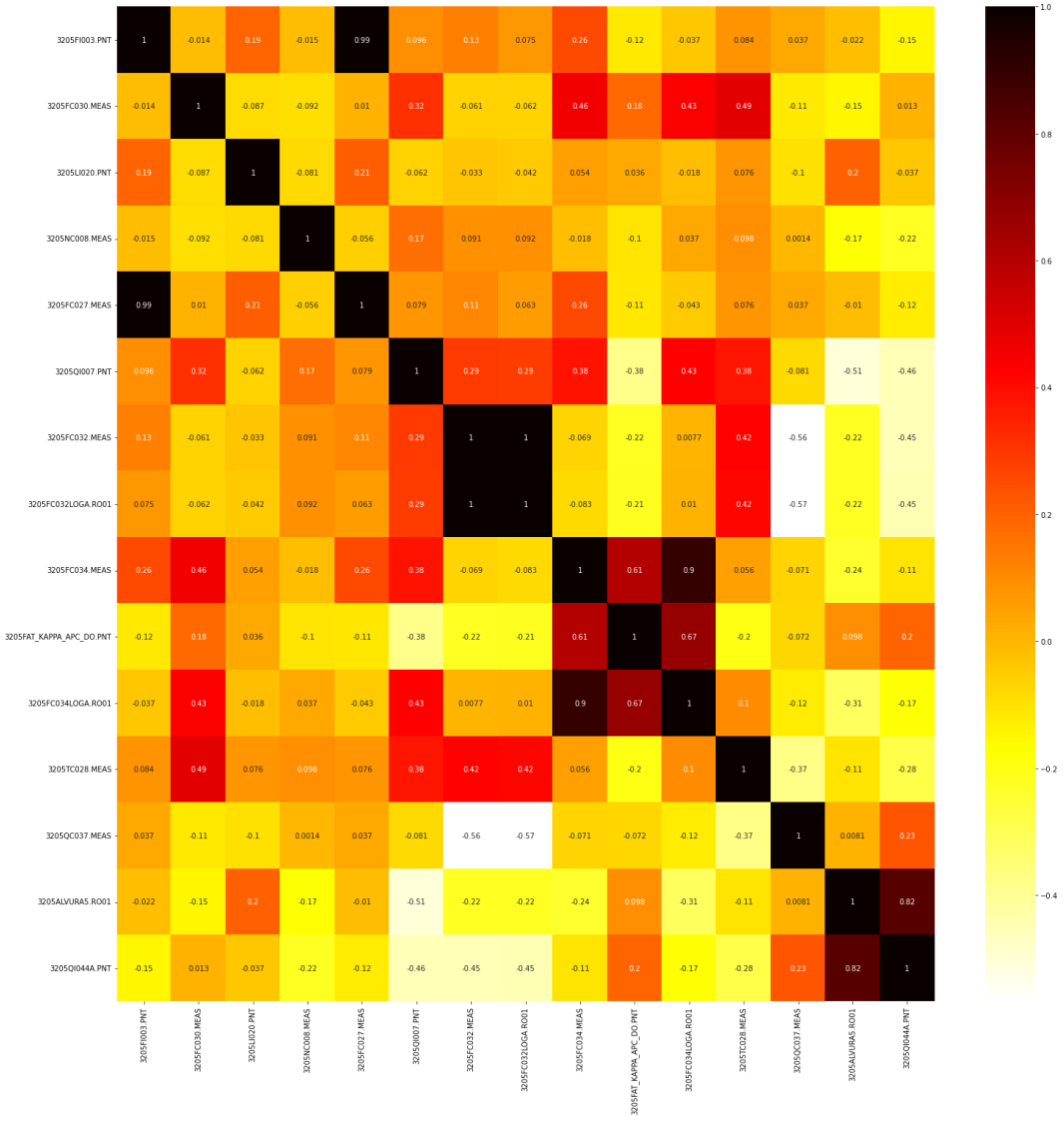
$$FIQ = Q3 - Q1 \qquad (05)$$

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

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

Utilizou-se novamente o recurso de mapa de calor, ilustrado na Figura 6, para reavaliar a correlação entre os atributos e a variável a ser predita, com o intuito de verificar se houve melhora nas correlações. O mesmo foi feito para as medidas de momentos de obliquidade e curtose. As equações para remoção dos *outliers* foram:

Figura 6 – Mapa de calor entre variáveis de entrada e variável a ser predita.



Fonte: Próprio autor (2022).

Uma vez tendo os dados extraídos, tratados e pré-processados, foi possível a utilização de métodos de aprendizado de máquina para extração de padrões de comportamento do atributo a ser predito. Para tal, foram utilizados os seguintes algoritmos de aprendizado de máquina supervisionado: *Decision Tree*, *Random Forest*, *XGBoost* e *lightGBM*. Em seguida utilizaram-se as métricas de acurácia, *Mean Absolute Error* (*MAE*), *Mean Squared Error* (*MSE*) e *Root Mean Squared Error* (*RMSE*) para avaliação e comparação dos resultados obtidos pelos algoritmos.

Para a aplicação dos algoritmos, inicialmente segmentou-se os dados de forma que o atributo a ser predito fosse categorizado como variável de saída (y) e os demais atributos como variáveis de entrada (**x**). Em seguida, após a segmentação, os *datasets* foram separados em dois conjuntos de igual tamanho, sendo o primeiro utilizado para ajuste dos parâmetros (conjunto de treino) e o segundo para testar a acurácia e generalização dos modelos em dados não conhecidos (conjunto de teste). Também foi utilizado o algoritmo de validação cruzada *K-fold* nos dados de teste, para avaliar o poder de generalização dos algoritmos em dados não conhecidos (conjunto de teste), fazendo o uso do coeficiente de determinação (R^2) médio do número de partições dos dados (*folds*) escolhidos.

O método de ajuste de todos os algoritmos utilizados no trabalho se dá com a formação de dois *datasets* após filtragem dos *outliers*, *dataset* de treino e de teste, execução do algoritmo selecionado sem ajuste de hiperparâmetros (valores padrão) e em seguida de posse dos indicadores de acurácia e poder de generalização obtidos nos dados de treinamento, foi realizada a sintonia dos hiperparâmetros baseada em tentativa e erro buscando o melhor ajuste das métricas obtidas nos dados de treinamento, ficando o *dataset* de teste exclusivo para medir a real performance do modelo, uma vez que estes dados não são conhecidos por ele.

5 RESULTADOS E DISCUSSÃO

5.1 Modelagem e sintonia

Os resultados das métricas de acurácia e poder de generalização obtidos a partir dos dados de treinamento foram utilizados para ajustes dos hiperparâmetros em todos os algoritmos, evitando-se assim o ajuste dos hiperparâmetros baseado nos resultados das métricas obtidas nos dados de teste, o que poderia caracterizar vazamento de dados. Como consequência disso os modelos tenderiam a super ajustar (*overfitting*).

O primeiro algoritmo utilizado para virtualizar a medição de alvura foi o regressor *Decision Tree Regressor* da biblioteca *sklearn*, fazendo uso dos hiperparâmetros *max_depth*, *min_samples_leaf* e *random_state*, sendo este último fixado em 1 para facilitar a reprodutibilidade dos diferentes testes submetidos com o uso de diferentes sintonias dos hiperparâmetros.

O hiperparâmetro *min_samples_leaf* define o número mínimo de amostras necessárias para formar um nó folha, o seu uso adequado mitiga a formação de nós folha com valores destoantes (*outliers*), reduzindo as chances do modelo super ajustar (*overfitting*). Outro hiperparâmetro bastante utilizado neste tipo de algoritmo, com o objetivo de limitar o crescimento de ramos da árvore (nível de profundidade) é o *max_depth*. Os resultados dos ajustes são apresentados na Tabela 2.

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

Algoritmo	**Hiperparâmetro**	**Valor**	**Métricas de acurácia**				**Comentários**
			MAE	***MSE***	***RMSE***	***R²***	
Decision Tree	*min_samples_leaf* *max_depth*	1 3	0,52758	0,43117	0,65664	0,70697	*Underfit*
	min_samples_leaf *max_depth*	1 *none*	0,00000	0,00000	0,00000	1,00000	*Overfit* (parâmetros padrão)
	min_samples_leaf *max_depth*	500 *none*	0,41993	0,29232	0,54067	0,80133	*Good fit*

Fonte: Próprio autor (2022).

Utilizando os hiperparâmetros padrão (*default*) observou-se a ocorrência de *overfitting*, uma vez que os indicadores de acurácia e poder de generalização obtiveram o melhor valor possível nos dados de treinamento, o que normalmente provoca resultados ruins em dados não conhecidos. Dando prosseguimento na sintonia dos hiperparâmetros, buscou-se fazer uso do

parâmetro *max_depth* com o intuito de reduzir a profundidade da árvore, reduzindo assim sua complexidade e consequentemente reduzindo significativamente o *overfitting* através da redução do R^2, tentando manter o indicador de acurácia *RMSE* dentro das especificações de projeto de 0,3%.

Com este ajuste realizado, tanto o poder de generalização quanto os indicadores de acurácia ficaram muito fora das metas estabelecidas de $R^2 > 80\%$ e $RMSE < 0{,}3\%$, motivando outras tentativas de sintonia. Optou-se na utilização do hiperparâmetro *min_samples_leaf* que também possui a característica de reduzir a complexidade da árvore através do aumento do número de amostras para a formação de um nó folha. Essa estratégia levou a um bom ajuste porém ainda fora das especificações de projeto fazendo-se necessário a utilização de algoritmos com mais recursos.

O próximo algoritmo escolhido foi a *Random Forest*, o qual utiliza média da previsão de várias *Decision Trees* (*ensemble*) usando aleatoriedade. Inicialmente foi executado a *Random Forest* sem ajustes de hiperparâmetros (valores padrão) e posteriormente foram sintonizados o *n_estimators* e o *max_depth* obtendo os resultados ilustrados na Tabela 3.

Tabela 3 – Ajuste do hiperparâmetro *n_estimators* e *max_depth* em *Random Forest* dados de treino.

Algoritmo	**Hiperparâmetro**	**Valor**	**Métricas de acurácia**				**Comentários**
			MAE	***MSE***	***RMSE***	***R²***	
Random Forest	*n_estimators* *max_depth*	100 2	0,60082	0,54691	0,73953	0,62831	*Underfit*
	n_estimators *max_depth*	100 *none*	0,05446	0,00696	0,08340	0,99527	*Overfit*
	n_estimators *max_depth*	300 5	0,41126	0,27425	0,52369	0,81361	*Good fit*

Fonte: Próprio autor (2022).

Houve uma pequena melhora dos indicadores de acurácia com o uso do algoritmo *Random Forest*, em relação a *Decision Tree* simples, sendo que o indicador *RMSE* ainda não atingiu a meta estabelecida neste projeto sem ocorrência de sobre ajuste (*overfitting*). Optou-se ainda em utilizar um outro algoritmo para este tipo de problema, o *XGBoost*, objetivando o atingimento da meta do *RMSE* através da aplicação da técnica de *boosting*, ou seja, treinamentos sucessíveis baseados na derivada do erro (*gradient*) das árvores anteriores. Foi aplicado o algoritmo *XGBoost Regressor* da biblioteca *Sklearn*, fazendo uso dos hiperparâmetros *n_estimators*, *max_depth* e *learning_rate* para ajuste do modelo, obtendo-se os resultados da Tabela 4.

Tabela 4 – Ajuste do hiperparâmetro *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	0,31295	0,16616	0,40763	0,88707	Parâmetros padrão
	n_estimators *max_depth* *learning_rate*	375 10 0,05	0,08431	0,01389	0,11787	0,99056	*Overfit*
	n_estimators *max_depth* *learning_rate*	85 5 0,05	0,96908	1,09584	1,04682	0,25525	*Underfit*
	n_estimators *max_depth* *learning_rate*	570 5 0,05	0,19818	0,07030	0,26515	0,95222	*Good fit*

Fonte: Próprio autor (2022).

Houve uma melhora com o uso do *XGBoost* em relação ao algoritmo utilizado anteriormente, atingindo as metas estabelecidas do projeto para os indicadores *RMSE* e *R²*. Com a sintonia padrão este algoritmo foi mais robusto ao *overfitting* que os anteriores, porém ao permitir uma maior profundidade da árvore, também foi possível verificar a ocorrência deste.

Buscando-se mais uma vez a melhora do nível de acurácia, utilizou-se o *LightGBM*, desenvolvido pela *Microsoft*, caracterizado por obter resultados tão bons quanto o *XGBoost*, porém, com um tempo de execução significativamente inferior. O algoritmo *LightGBM Regressor* da biblioteca *LightGBM* foi implementado fazendo uso dos hiperparâmetros *n_estimators*, *max_depth*, *learning_rate* e *num_leaves* para ajuste do modelo, obtendo-se os resultados ilustrados na Tabela 5.

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

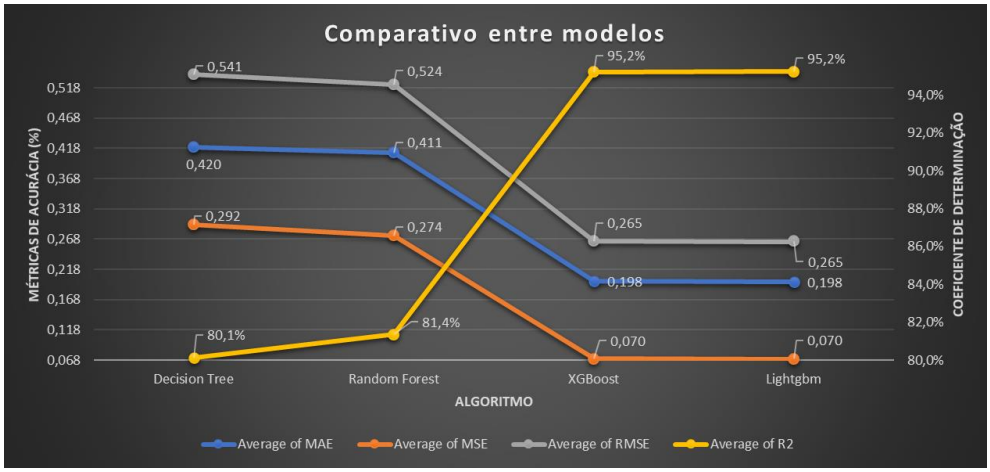
Modelo	Hiperparâmetro	Valor	Métricas de acurácia				Comentários
			MAE	***MSE***	***RMSE***	***R²***	
LightGBM	*num_leaves* *learning_rate* *n_estimators* *max_depth*	31 0,1 100 *none*	0,22027	0,08529	0,29205	0,94203	Parâmetros padrão
	num_leaves *learning_rate* *n_estimators* *max_depth*	60 0,15 500 10	0,08346	0,01281	0,11321	0,99129	*Overfit*
	num_leaves *learning_rate* *n_estimators* *max_depth*	2 0,15 10 5	0,64739	0,65622	0,81008	0,55402	*Underfit*
	num_leaves *learning_rate* *n_estimators* *max_depth*	20 0,15 248 5	0,19752	0,07009	0,26474	0,95237	*Good fit*

Fonte: Próprio autor (2022).

O algoritmo *LightGBM* obteve um resultado ligeiramente melhor em comparação com o algoritmo *XGBoost* nos indicadores *RMSE* e R^2. Assim como o XGBoost, este possuiu uma boa robustez ao *overfitting* com os hiperparâmetros padrão, além de ter sido possível observar a ocorrência dele com o aumento da quantidade de nós folha na árvore, o que permite com que nós folhas sejam criados com menor número de amostras e assim valores destoantes acabam impactando mais o resultado do *ensemble*.

A partir dos resultados obtidos de cada algoritmo executado, registrou-se o melhor resultado obtido de cada métrica de acurácia, bem como coeficiente de determinação ilustrados na Figura 7. É perceptível uma grande melhora na acurácia (redução do erro) e no coeficiente de determinação (aumento do R^2), ao comparar uma *Decision Tree* simples para *Random Forest* e a continuidade da melhora dos indicadores com a utilização dos algoritmos *LightGBM* e *XGBoost*.

Figura 7 – Métricas de acurácia por algoritmos dados de treino.



Fonte: Próprio autor (2022).

5.2 Validação com os dados de testes

Uma vez tendo os resultados das métricas anteriores, foram submetidos os algoritmos com os hiperparâmetros do melhor resultado encontrado, ao *dataset* de teste obtido através do *split* anteriormente realizado. Dessa forma é possível avaliar os modelos em dados não conhecidos conforme os resultados apresentados na Tabela 6.

Tabela 6 – Comparativo dos indicadores de acurácia e poder de generalização entre treino e teste.

Modelo	Hiperparâmetro	Valor	*Dataset*	*MAE*	*MSE*	*RMSE*	R^2
Decision Tree Regressor	*min_samples_leaf*	500	Treino	0,41993	0,29232	0,54067	0,80133
	max_depth	*none*	Teste	0,42378	0,29763	0,54555	0,79902
Random Forest Regressor	*n_estimators*	300	Treino	0,41126	0,27425	0,52369	0,81361
	max_depth	5	Teste	0,41473	0,27871	0,52793	0,81180
XGBoost Regressor	*n_estimators*	570	Treino	0,19818	0,07030	0,26515	0,95222
	max_depth	5	Teste	0,21046	0,08222	0,28674	0,94448
	learning_rate	0,05					
LightGBM Regressor	*n_estimators*	248	Treino	0,19752	0,07009	0,26474	0,95237
	max_depth	5					
	learning_rate	0,15	Teste	0,22127	0,08974	0,29957	0,93940
	num_leaves	20					

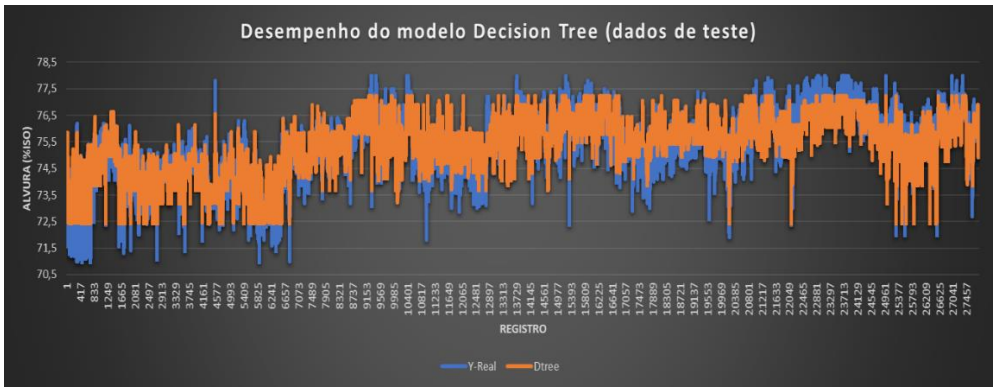
Fonte: Próprio autor (2022).

Conforme ilustrado na Tabela 6 é possível comparar os indicadores de acurácia *RMSE* e coeficiente de determinação R^2 entre os dados de treinamento e teste. Foi observado em todos os algoritmos um decréscimo já esperado no desempenho dos modelos nos dados de teste em comparação aos dados de treinamento. Os algoritmos de *Decision Tree* e *Random Forest* não atenderam à meta do projeto nos dados de teste. Já o *XGBoost* e o *LightGBM* conseguiram atingir a meta de acurácia com um *RMSE* menor do que 0,3% e um R^2 superior a 80%.

É esperado um pior desempenho dos indicadores nos dados de teste que pode ser causado por sobre ajuste (*overfitting*) ou sub ajuste (*underfitting*), ocorrido durante o processo de treinamento dos algoritmos, bem como em função de oportunidades de melhor filtragem dos atributos de entrada do modelo reduzindo assim *outliers* ainda remanescentes. Mesmo com a penalização observada ainda foi possível atingir as metas de *RMSE* inferior a 0,3% e R^2 superior a 80% nos algoritmos *XGBoost Regressor* e *LightGBM Regressor*.

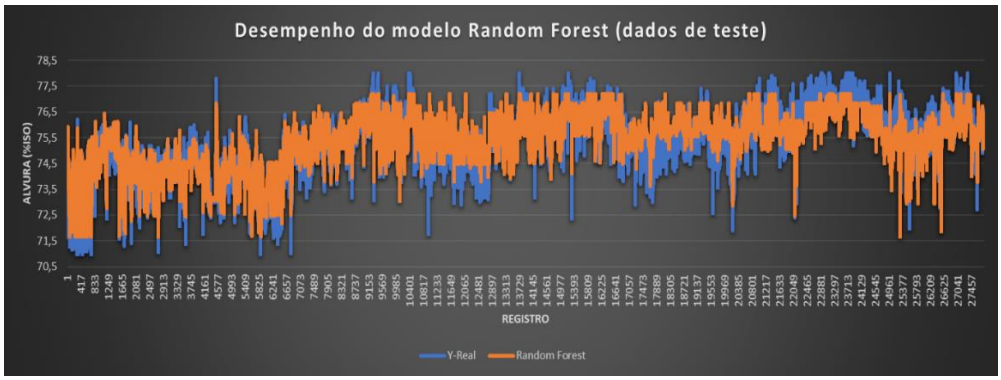
Os algoritmos *Random Forest Regressor* e *Decision Tree Regressor* não conseguiram atingir as metas tornando-se inviáveis para uso. Nas Figuras 8, 9, 10 e 11 está ilustrado o comparativo do desempenho da predição de cada algoritmo em relação aos dados reais fornecidos pelo robô analisador (dados de teste). A sobreposição da previsão de cada algoritmo em relação ao conjunto de teste (dados não conhecidos) demonstra de forma visual a performance obtida.

Figura 8 – Predição x Real - *Decision Tree* nos dados de teste.



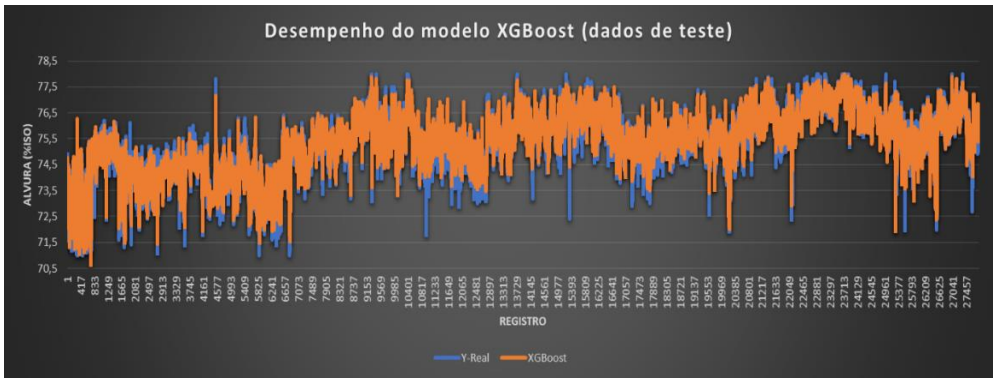
Fonte: Próprio autor (2022).

Figura 9 – Predição x Real – *Random Forest* nos dados de teste.



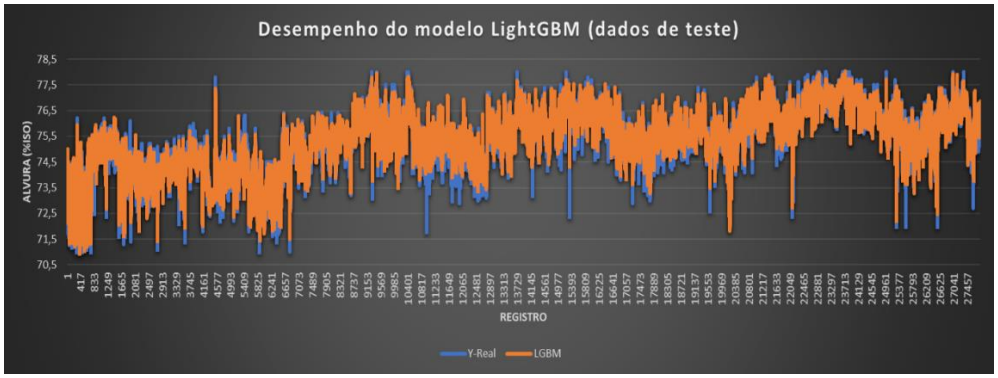
Fonte: Próprio autor (2022).

Figura 10 – Predição x Real – *XGBoost* nos dados de teste.



Fonte: Próprio autor (2022).

Figura 11 – Predição x Real – *LightGBM* nos dados de teste.



Fonte: Próprio autor (2022).

5.3 Validação cruzada *K-fold*

Para avaliar a capacidade de generalização dos modelos gerados por cada algoritmo sobre os dados de teste, utilizou-se o método de validação cruzada denominado *K-fold*, da biblioteca *Sklearn*. Os parâmetros ajustados durante este processo de validação cruzada foram *n_split* (número de *folds*), *shuffle* (embaralhar) definido como verdadeiro para misturar os subconjuntos e fixado o random_state (estado de aleatoriedade) para termos repetibilidade dos experimentos. Manipulando-se o número de *folds*, chegou-se aos resultados de cada algoritmo, apresentados na Tabela 7.

Tabela 7 – Validação cruzada *K-fold* nos dados de teste.

Validação Cruzada *K-fold*		***Decision Tree Regressor***								
Hiperparâmetro		***n_splits***	***random_state***	***shuffle***	***n_splits***	***random_state***	***shuffle***	***n_splits***	***random_state***	***shuffle***
Valor		**3**	**7**	***True***	**5**	**7**	***True***	**10**	**7**	***True***
Decision Tree	**Média**		0,77990			0,78924			0,79372	
	Desvio Padrão		0,003214			0,00650			0,00718	
Random Forest	**Média**		0,80941			0,80816			0,80783	
	Desvio Padrão		0,00450			0,00858			0,00936	
XGBoost	**Média**		0,93609			0,93742			0,93784	
	Desvio Padrão		0,00281			0,00233			0,00246	
LightGBM	**Média**		0,9370717			0,93858			0,93913	
	Desvio Padrão		0,00254			0,00219			0,00264	

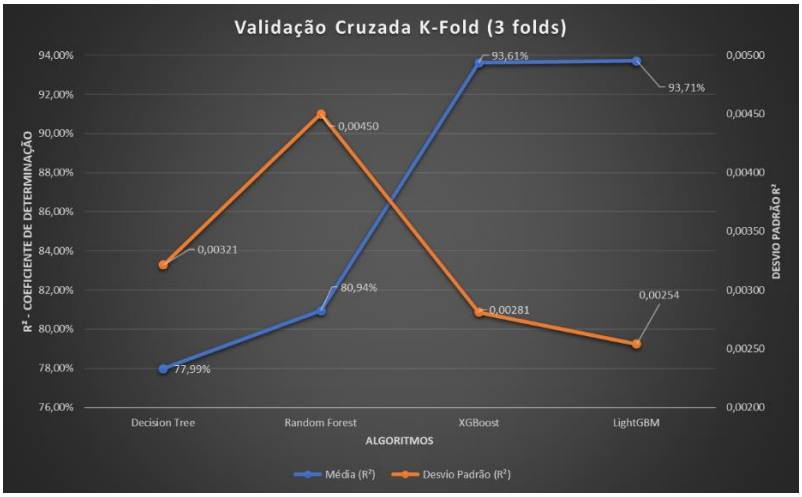
Fonte: Próprio autor (2022).

É possível observar, avaliando a média dos coeficientes de determinação bem como seu desvio padrão, que bons resultados foram geralmente alcançados, com excessão do algoritmo *Decision Tree* que não atingiu o valor mínimo da meta estabelecida R^2 igual a 80%, conforme ilustrado nas figuras 12, 14 e 16. A *Random Forest* apesar de ter atingido a meta estabelecida tomando-se como base a média entre os *folds*, não conseguiu atingir a meta em alguns deles como ilustrado na figura 17, além de ter sido o algoritmo de maior desvio padrão médio. Os

algoritmos *XGBoost* e *LightGBM* alcançaram desempenho similares, ambos superando a meta estabelecida bem como sendo melhores que os algoritmos *Descision Tree* e *Random Forest*.

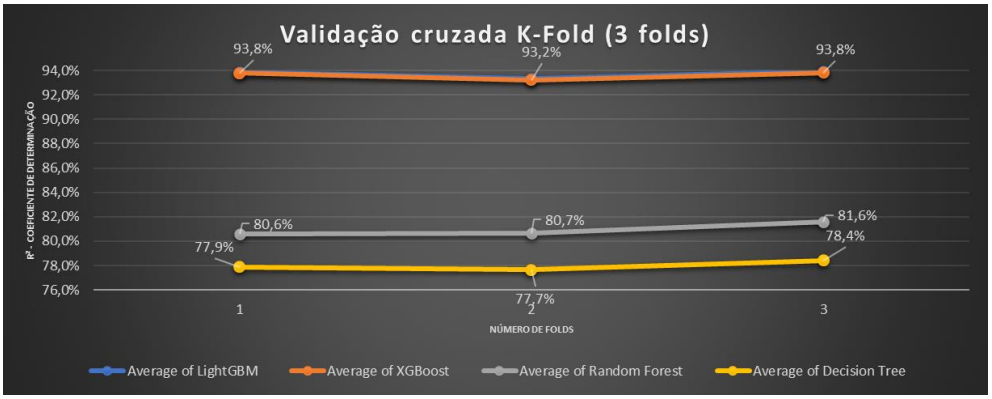
De posse dos resultados de validação cruzada *K-fold* com 3, 5 e 10 *folds*, aparentemente, todos os modelos desempenharam bem em relação ao poder de generalização, possuindo desvio padrão relativamente baixo, mantendo um coeficiente de determinação alto, conforme ilustrados nas figuras 12 a 17.

Figura 12 – *K-fold* (3 *folds*) média e desvio padrão R^2 por algoritmo dados de teste.



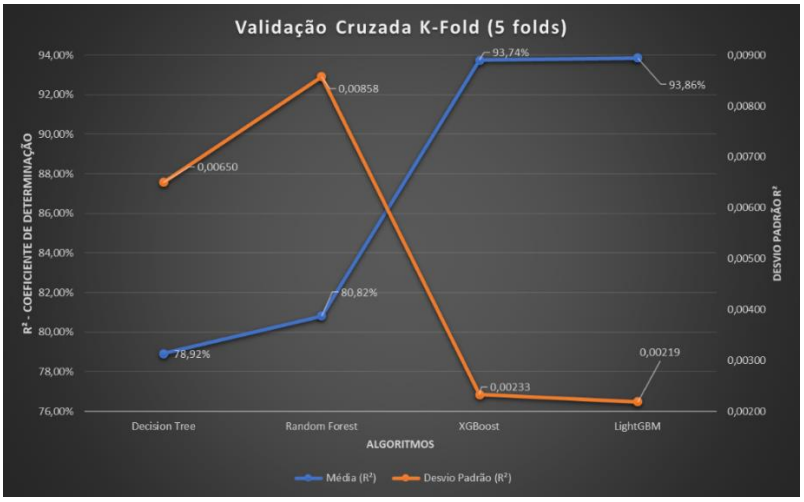
Fonte: Próprio autor (2022).

Figura 13 – *K-fold* (3 *folds*) coeficiente de determinação (R^2) por algoritmo dados de teste.



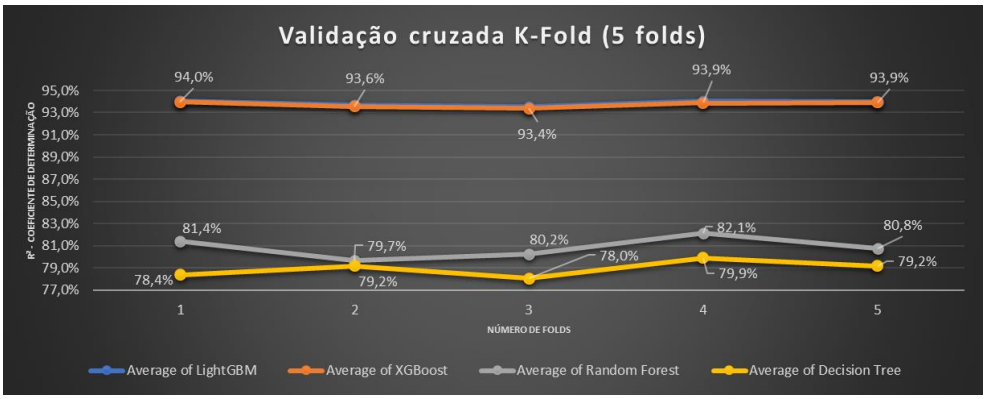
Fonte: Próprio autor (2022).

Figura 14 – *K-fold* (5 *folds*) média e desvio padrão *R²* por algoritmo dados de teste.



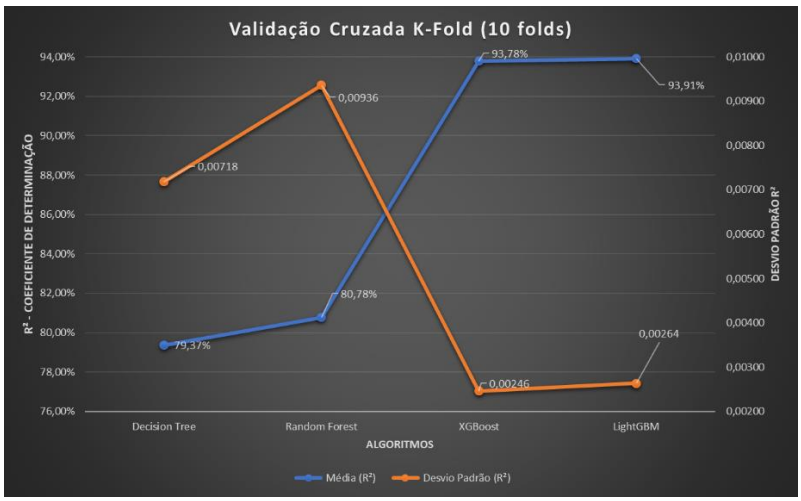
Fonte: Próprio autor (2022).

Figura 15 – *K-fold* (5 *folds*) coeficiente de determinação (*R²*) por algoritmo dados de teste.



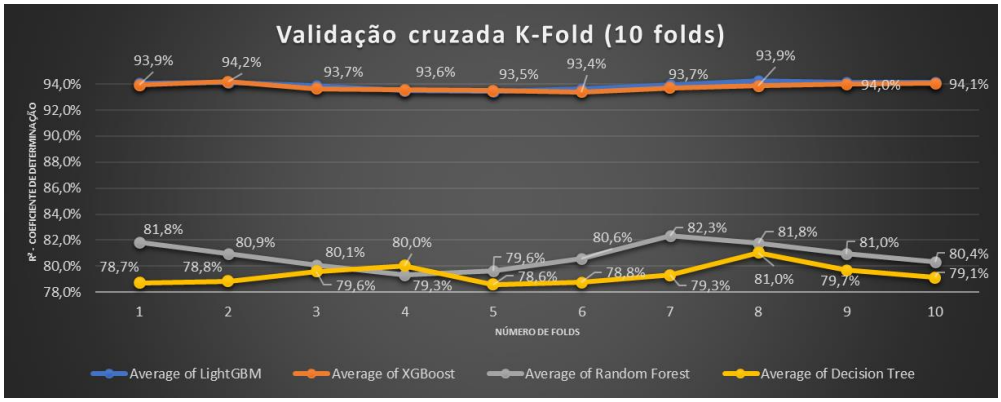
Fonte: Próprio autor (2022).

Figura 16 – *K-fold* (10 *folds*) média e desvio padrão *R²* por algoritmo dados de teste.



Fonte: Próprio autor (2022).

Figura 17 – *K-fold* (10 *folds*) coeficiente de determinação (R^2) por algoritmo dados de teste.



Fonte: Próprio autor (2022).

6 CONCLUSÃO

Respeitar as etapas de pré-processamento, extração de padrões e pós processamento na construção do modelo de predição é essencial para obtenção de bons resultados e o emprego de técnicas para a tratativa dos dados contidos no *dataset* pode elevar os indicadores de acurácia e poder de generalização, como por exemplo o uso de *PCA*.

Com os resultados obtidos através dos indicadores de acurácia e poder de generalização pode-se afirmar que é possível utilizar algoritmos de AM, como descritos neste trabalho para a construção de um sensor virtual de alvura que permite otimizar o processo de branqueamento. Dentre os algoritmos utilizados os que representaram melhor desempenho foram o *XGBoost Regressor* e o *LightGBM Regressor*, uma vez que ambos atenderam às especificações de projeto.

Torna-se assim viável o emprego deste recurso de utilização de AM de forma abrangente nos mais diversos processos da indústria de papel e celulose podendo vir a trazer grandes impactos em produção, redução de custos variáveis, com impactos também em meio ambiente e segurança. Uma vez tendo obtido êxito no desenvolvimento de um sensor virtual de alvura para o primeiro estágio do Branqueamento, torna-se viável criar algoritmos de AM para os demais estágios com essa mesma finalidade de predição de alvura.

O controle de alvura que atualmente é *feedback* poderá ser melhorado com a implementação de novas estratégias antecipativas (*feedforward*), que somadas com a estratégia atual, farão com que ocorram mais ações nos atuadores promovendo uma otimização de reagentes químicos, culminando na redução da variabilidade da alvura e consequentemente permitindo ao operador trabalhar mais próximo dos limites de especificação, reduzindo assim o consumo de químicos, mantendo a qualidade final do produto dentro das especificações.

Para aumentar a acurácia e poder de generalização pode ser feito uso de novos algoritmos de AM, como por exemplo, as redes neurais artificiais. Outros atributos estratégicos para o processo de produção de celulose tanto da área de Branqueamento quanto das demais áreas também são candidatos a serem virtualizados buscando a otimização de outros processos. O aprendizado obtido também poderá ser utilizado no emprego de criação de sensores virtuais com o foco em disponibilidade de ativos e predição de falhas.

REFERÊNCIAS

[1] Jardim, C. M. (2010). **Impactos de modificações físico-químicas das fibras de eucalipto na qualidade da polpa branqueada.** 191 f. Tese (Doutorado em Manejo Florestal; Meio Ambiente e Conservação da Natureza; Silvicultura; Tecnologia e Utilização) - Universidade Federal de Viçosa, Viçosa, 2010.

[2] 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.

[3] Smook, G. A. (1989). **Overview of the pulp and paper industry from a chemical industry perspective**. Publihser: Journal of Chemical Technology & Biotechnology, 15-27.

[4] Rabelo, M. S., Silva, V. L., Barros, D. P. D., Colodette, J. L., Sacon, V. M., & Silva, M. R. D. (2009). **Branqueamento de polpa celulósica kraft de eucalipto com peróxido ácido ativado por molibdênio**. Publisher: Química Nova, 1095-1098.

[5] Perdigão, P. R.; Andrade, M. C. (2004) **Manual de Operação Color Touch II**. Publisher: Modelo ISO, CETEM, Rio de Janeiro, 3-9.

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

[7] Han, J., Pei, J., & Kamber, M. (2011). **Data mining: concepts and techniques**. Elsevier. Publisher: Morgan Kaufmann, (2), 35-45.

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

[12] 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.

[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] Gomes, L. V. (2021). **Aplicação de metodologias estatísticas e machine learning para validação de analisadores contínuos para indústria de celulose**. Tese (Graduação) – Escola de Engenharia Química, Universidade Federal do Pampa (Unipampa), Bagé, 2021.

[15] Domingues, M. A. (2019). **Desenvolvimento de um sensor virtual para controle da resistência à tração do papel em uma planta de polpa CTMP**. Tese (Pós-graduação) – Escola de Engenharia e Ciência dos Materiais, Setor de tecnologia, Universidade Federal do Paraná, Curitiba, 2019.

[16] Cavanha Filho, A. O. (1996). **Estatística Básica**. Publisher: *Qualitymark*, 42.