11º Congresso Brasileiro de Engenharia de Fabricação
24 a 26 de maio de 2021, Curitiba, PR, Brasil

UTILIZAÇÃO DE ALGORITMOS DE MACHINE LEARNING PARA CRIAÇÃO DE MODELO PREDITIVO DE FALHAS EM MÁQUINA CNC

Gianfranco Daroit Cavassin
Flavia Ayumi Ushiwata
Renato Fernandes Grottoli
Fernando Deschamps
Universidade Federal do Paraná, DEMEC, Av. Cel. Francisco H. dos Santos, 210 CEP 81531-990, Curitiba /PR.
gianfrancocavassin@ufpr.br
flavia.ayumi@ufpr.br
renato.grottoli@upfr.br
fernando.deschamps@ufpr.br

Resumo. *No mundo atual da manufatura, o papel da manutenção tem recebido cada vez mais atenção pois quando bem executada, pode ser um fator estratégico para atingir os objetivos corporativos. No contexto da indústria 4.0 e com o aumento exponencial da quantidade de dados extraídos dos processos de manufatura, as últimas tendências apontam para a utilização de uma abordagem preditiva, visando a execução de manutenções baseadas em modelos matemáticos. Nesse cenário, algoritmos e métodos em Machine Learning vem emergindo como uma ferramenta promissora para aplicações de manutenção preditiva para a prevenção de falhas em equipamentos que compõem as linhas de produção. Esse trabalho tem como objetivo a comparação e posterior análise de quatro algoritmos de machine learning para a identificação de falha na máquina. Em uma primeira etapa, os principais conceitos envolvidos são apresentados. Em seguida, são discutidos os métodos e parâmetros utilizados para a criação e avaliação dos modelos. Por fim, cada um dos algoritmos criados são analisados separadamente e então comparados visando trazer uma possível solução para a predição de falhas e aplicações em um caso prático de uma máquina CNC.*

Palavras chave*: Machine Learning, manutenção preditiva, algoritmo, predição de falhas*

1. INTRODUÇÃO

Atualmente, a indústria está passando pelo que especialistas chamam de "A Quarta Revolução Industrial", também chamada de indústria 4.0. Este fato está fortemente relacionado com a integração entre os sistemas físicos e digitais de meios de produção. A integração entre esses ambientes permite a coleta de grande quantidade de dados por diferentes equipamentos, localizados em diferentes setores e ambientes da fábrica e linha de produção (Carvalho, *et al.* 2019).

As mudanças decorrentes dessa transformação digital se estendem também ao ambiente da manufatura e manutenção, de modo que uma das principais abordagens provenientes dessa transformação é a Manutenção Preditiva (PdM, do inglês *Predictive Maintenance).* A ideia básica da manutenção preditiva é de monitorar e analisar um sistema em tempo real visando acionar uma manutenção de maneira proativa e prevenir uma quebra total do sistema (Zenisek, e al. 2019). A predição é o tipo mais eficiente e sua aplicação se beneficia desta revolução. Enquanto uma manutenção corretiva implica na correção de uma falha no momento da quebra e uma manutenção preventiva implica em cronogramas de manutenções empiricamente definidos, a manutenção preditiva se baseia na real condição da máquina (Zenisek, *et al*. 2019).

Ainda nesse contexto, métodos de Machine Learning (ML) tem emergido como uma ferramenta promissora para aplicações em manutenções preditivas para a detecção e prevenção de falhas dentro da linha de produção. Entretanto, a performance dessa aplicação depende da escolha apropriada do algoritmo (Carvalho, *et al*. 2019) e dos parâmetros utilizados. Dentre estes a vibração tem se destacado não só como forma de monitoramento mas como um dos principais indicadores para a predição de falhas como apresentados nos trabalhos de Wu, *et al.* (2017) e Zhang, *et al*. (2019).

O objetivo do presente trabalho é justamente a aplicação de algoritmos de machine learning, visando a construção de modelos preditivos com o objetivo de realizar a predição de falhas. Para tanto, foram disponibilizados históricos de manutenção e vibração de uma máquina CNC GROB BZ 500, que serão usados como base para o treinamento do algoritmo e a construção e avaliação do modelo. Serão selecionados algoritmos já presentes na literatura, tais como, regressão logística (RL), rede neural (RN), árvore de decisão (AD), e floresta de decisão (DF). Em seguida será realizada uma avaliação individual e comparativa de cada um dos algoritmos visando a definição do melhor modelo para o problema proposto.

2. REVISÃO BIBLIOGRÁFICA

Nesta seção serão revisados os conceitos e o referencial teórico para entendimento do uso de alguns algoritmos mais comuns na manutenção preditiva e práticas aplicadas a tratamentos de dados e modelagem no aprendizado de máquina. Notamos que existem vários estudos na literatura que abordam esses temas. Segundo o trabalho de Zhang, *et al.* (2019), o ML pode atender às necessidades do PdM de maneira muito significativa. Em suas conclusões, a precisão média de predição na literatura revisada atingiu 95,06%, e a maior precisão obtida foi de 100%.

2.1. Comparação de algoritmos

Entre os vários métodos de aprendizagem disponíveis, o mais utilizado, segundo Braga, *et al.* (2000), é o método de aprendizagem supervisionada. Este método consiste em um recurso em que entradas e saídas são fornecidas por um agente externo, isto é, um histórico de dados rotulados que serve de parâmetro para organizar os pares ordenados de entrada e saída. Na "Tabela 1", é apresentado um comparativo dos principais tipos de algoritmos de classificação. Posteriormente cada um deles será tratado de maneira sucinta.

Tabela 1. Comparativo entre algoritmos de Machine Learning

Família	Algoritmo	Tipo	Precisão	Tempo de treinamento	Linearidade	Forma de aprendizado
Classificação	Regressão logística	Duas classes	Satisfatório	Rápido	Sim	Supervisionado
Classificação	Floresta de decisão	Duas classes	Excelente	Moderado	Não	Supervisionado
Classificação	Árvore de decisão aumentada	Duas classes	Excelente	Moderado	Não	Supervisionado
Classificação	Rede neural artificial	Duas classes	Satisfatório	Moderado	Não	Supervisionado

2.1.1. Árvore de decisão aumentada *(Boosted Decision Tree)*

Em uma árvore de decisão tradicional, conhecida como *Decision Tree*, a estrutura do classificador é organizada no formato de uma árvore. Cada nó intermediário, não folha, da árvore corresponde a uma característica e um condicional aplicado a mesma (Nogare e Zavaschi, 2016). Os algoritmos de árvore de decisão dividem os dados em dois ou mais conjuntos homogêneos. Eles usam as regras if–then para separar os dados com base no diferenciador mais significativo entre os pontos de dados. A "Figura 1" ilustra um esquemático desse método.

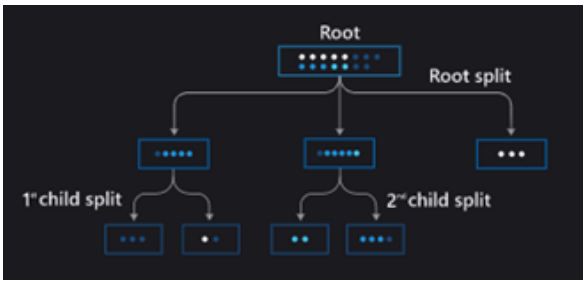

Figura 1. Esquema Árvore de Decisão. Fonte: https://azure.microsoft.com/overview/machine-learning-algorithms/

Uma árvore de decisão aumentada, Boosted Decision Tree, é um método de aprendizagem de máquina supervisionado baseado em um ensemble, ou seja, um conjunto de classificadores. Neste caso são utilizadas várias árvores onde a segunda corrige os erros da primeira, a terceira corrige os erros da segunda e da primeira e assim por diante. A predição ocorre com a utilização de todas as árvores juntas no ensemble. (Nogare e Zavaschi, 2016)

2.1.2. Rede neural artificial *(Artificial Neural Network)*

A rede neural artificial se trata de um artifício computacional que possui o objetivo de armazenar conhecimento experimental, estimar dados e o tornar disponível para utilização e resolução de problemas. Esse tipo de análise tem a capacidade de, por exemplo, utilizar o histórico de variáveis temporais, modelando-os com o intuito de prever seu comportamento futuro. (Berry e Linoff, 2004).

O neurônio também pode ser descrito pelo modelo matemático de generalização, conforme a "Eq. 1":

$$yk=F(\Sigma(xiwki)+bkni=1) \quad (1)$$

Em um neurônio comum existe a interação entre os dados de entrada (x1, x2, ..., xi), e os pesos sinápticos (wk1, wk2, …, wki). Esses pesos possuem a função de ponderar as entradas. Quanto maior o peso sináptico do dado, maior a sua influência dentro da rede. Após este primeiro processamento, os dados ponderados são somados em uma única saída uk. Para a saída final de informações yk, os dados são somados ao bias bk, uma constante que auxilia o modelo a se adequar aos dados recebidos. Esses dados passam pela função de ativação φ, que transforma a saída do neurônio em uma função não linear (Haykin, 1999). Esse processo pode ser representado utilizando um diagrama de blocos conforme mostrado na "Fig. 2".

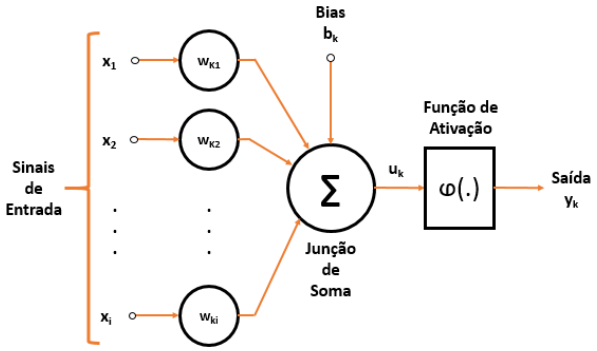

Figura 2 - Diagrama de blocos do neurônio. Adaptado de Haykin (1999).

2.1.3. Regressão logística *(Logistic Regression)*

Segundo Nogare e Zavaschi (2016), o que este modelo busca é a probabilidade de algum evento acontecer ou não. As variáveis independentes serão responsáveis por determinar o resultado que terá a variável dependente, que nesse caso, será binária, valores discretos, e não valores contínuos. Para simplificar esta etapa, o resultado que se busca é um valor normalizado entre 0 e 1, e a função logística ajuda a definir se a classificação binária provável é de 0 ou de 1.

A equação padrão para uma regressão logística pode ser representada pela "Eq. 2" a seguir:

$$h\theta(x) = g(\theta\, Tx) \tag{2}$$

Nessa equação, $h\theta(x)$ é a hipótese (que será o valor de Y, a variável dependente) em um determinado x. O resultado dessa equação genérica geralmente é apresentado por uma curva contínua em forma de S aos dados (sigmóide).

2.1.4. Floresta de decisão *(Decision forest)*

Em uma floresta de decisão, várias árvores de decisão são criadas e, em seguida, vota-se na classe de saída mais popular. A votação é um dos métodos mais conhecidos para gerar resultados em um modelo Ensemble. Cada árvore da floresta de decisão gera um histograma de frequência não normalizado dos rótulos. O processo de agregação soma esses histogramas e normaliza o resultado para obter as "probabilidades" para cada rótulo. As árvores que têm alta confiança de previsão terão um peso maior na decisão final do Ensemble. (Guia de IA do Azure para soluções de manutenção preditiva, último acesso: 13/03/2021)

2.2. Dados em desequilíbrio

Em problemas de classificação, se houver mais exemplos de uma classe que de outras, o conjunto de dados será considerado em desequilíbrio. Esse problema ocorre se uma classe corresponder a menos que 10% dos dados. A classe sub-representada é chamada de classe minoritária e geralmente se sobressai como dados mais relevantes que o resto, por isso o foco está no desempenho desta detecção. Prever incorretamente uma classe positiva como uma classe negativa pode ter custo maior do que o contrário, conhecida como perda diferente. Métodos de estímulo como a árvore de decisão aumentada geralmente mostram um bom desempenho para dados desbalanceados (Guia de IA do Azure para soluções de manutenção preditiva, último acesso: 13/03/2021).

No caso de desequilíbrio de classe, o desempenho da maioria dos algoritmos de aprendizado padrão fica comprometido, já que eles tentam minimizar a taxa de erro geral. Consequentemente, as métricas de avaliação convencional, como a precisão geral na taxa de erro, não são suficientes. Dois métodos mais comuns podem ajudar a tratar este problema:

- Técnicas de amostragem: A sobreamostragem replica casos de minoria para que a diferença quantitativa seja suavizada. A subamostragem aleatória corresponde à remoção de um conjunto de dados de treinamento da

classe que está em maioria. No entanto, esse sobreajuste/remoção pode causar perda de conceitos importantes dessa classe. Pode por fim ser feita uma amostragem híbrida, com uma combinação dos dois.

- Aprendizado sensível ao custo: fornece alta precisão de previsão para a classe em desvantagem, por meio da atribuição de um custo de classificação incorreta de classe minoritária para minimizar o custo geral.

Dentre os métodos de sobreamostragem mais conhecidos pode-se citar o SMOTE (técnica de sobreamostragem minoritária sintética). Esta técnica estatística consiste em criar observações intermediárias entre dados parecidos, sendo assim, as novas instâncias não são apenas cópias de casos minoritários existentes. Em vez disso, o algoritmo usa exemplos do espaço de recurso para cada classe de destino e seus vizinhos mais próximos. O algoritmo, em seguida, gera novos exemplos que combinam recursos do caso de destino com recursos de seus vizinhos. Essa abordagem aumenta os recursos disponíveis para cada classe e torna os exemplos mais gerais (Guia de IA do Azure para soluções de manutenção preditiva, último acesso: 13/03/2021).

3. MATERIAIS E MÉTODOS

Para o estudo de caso, uma máquina CNC com 5 eixos modelo GROB BZ 500, fabricada no ano 2000, foi selecionada dentre as disponíveis dentro da linha de montagem de uma empresa multinacional de grande porte na área de autopeças, localizada em Curitiba, PR, para a coleta dos dados necessários.

3.1 Ferramentas utilizadas (Microsoft Azure)

O estúdio Azure Machine Learning é um portal da Web para desenvolvedores cientistas de dados. O estúdio combina experiências sem e com código para obter uma plataforma de ciência de dados inclusiva. A parte do Machine Learning possui um ambiente baseado em nuvem que pode ser usado para treinar, implantar e gerenciar modelos.

A escolha dessa ferramenta é baseada principalmente nos seguintes motivos: (a) versão estudantil gratuita, (b) fácil compartilhamento do projeto, (c) solução sem a necessidade de codificação, e (d) grande diversidade de algoritmos de machine learning disponíveis para utilização.

3.2 Apresentação de parâmetros

Os dados disponibilizados para a análise de parâmetros e construção dos modelos preditivos podem ser divididos em dois grandes relatórios: (a) histórico de manutenções da máquina, e (b) dados de vibração da máquina. A base de dados (a) consiste nos dados referentes aos eventos de manutenção, ou ordens, dentre os períodos de janeiro de 2018 a março de 2020. A base de dados (b), por sua vez, corresponde ao histórico da vibração da máquina coletado entre os dias 24/11/2019 e 15/05/2020 pelo sensor montado próximo ao mancal traseiro. Os tipos de vibração coletados foram:

1. SE02_v_RMS_PROCESSO (m/s): Corresponde à média quadrática do valor de velocidade do movimento.
2. SE02_a_Peak_IMPACTO (m/s2): Corresponde à medição do pico da senóide da aceleração do movimento.
3. SE02_a_RMS (m/s2): Corresponde à média quadrática do valor de aceleração do movimento.
4. SE02_v_RMS_ISO 17243 (m/s): Corresponde à média quadrática do valor de velocidade do movimento. Esses valores são coletados sob demanda, no caso estudado a cada 40 ciclos.

Após o tratamento e análise dos relatórios obteve-se dados referentes ao período entre os dias 28/10/2019 e 15/05/2020, com os seguintes parâmetros: (a) data, (b) média dos dados de vibração do dia, (c) máximo valor da vibração captada no dia, (d) desvio padrão dos valores de vibração do dia, (e) média dos dados de vibração no período pré-falha, (f) número de manutenções planejadas no período pré-falha, (g) número de manutenções preventivas no período pré-falha. Todos os parâmetros de vibração apresentados (média, desvio padrão e máximo) são calculados para os 4 tipos de vibração apresentados. Por fim, foram totalizados 18 parâmetros diferentes com uma possível influência para os eventos de falha.

Os parâmetros referentes ao período pré-falha tem como objetivo analisar a influência de ocorrências e alterações do processo e da máquina de dias anteriores ao dia da falha para a ocorrência da mesma. Após discussão com especialistas da área, o período pré-falha é definido como o intervalo de 2 dias anteriores ao referido dia até o dia em questão. Dessa forma, as médias da vibração correspondem à média dos valores da vibração neste período de 2 dias e o número de manutenções no período pré-falha corresponde à contagem de manutenções neste mesmo período.

Visto que optou-se pela implementação de modelos binários de machine learning, a identificação da ocorrência de falhas foi realizada por meio da utilização de uma coluna denominada “FALHA”, onde “Sim” corresponde a uma falha e “Não” corresponde a dia sem o registro de falha.

3.4 Solução para desbalanceamento de dados

Dentro do presente trabalho, o desbalanceamento de dados se mostrou um empecilho visto que a grande maioria das entradas da amostra correspondiam a casos de não falha (cerca de 90%). Devido principalmente ao baixo volume de dados disponíveis, optou-se pela utilização do SMOTE, uma técnica de Oversampling, para a resolução desse problema.

Dentro da plataforma, aplicou-se como parâmetro uma porcentagem SMOTE de 800%. Após a utilização do método a proporção de casos de não falha passou a ser de aproximadamente 50%.

3.5 Método para construção e avaliação dos modelos preditivos

O método para a construção e avaliação dos modelos, consiste primeiramente na inserção dos inputs dentro da plataforma. Em seguida, a amostra passa por um processo de tratamento de dados faltantes, principalmente compostos por dados de vibração. No presente trabalho, optou-se por substituir os dados faltantes pela média entre os valores da coluna. O tratamento de dados é finalizado com a aplicação do método SMOTE.

Após o tratamento dos dados, a amostra é dividida aleatoriamente em uma proporção comumente utilizada de 75/25, de modo que 75% dos dados são utilizados para o treinamento do algoritmo e construção do modelo preditivo. Em seguida o modelo é aplicado dentro dos 25% restantes da amostra, visando comparar os resultados reais e aqueles obtidos pelo modelo. Por fim, o modelo é avaliado, trazendo uma visão de acurácia e porcentagem de verdadeiros positivos e negativos de acordo com o limite definido.

3.6 Parâmetros utilizados para a construção dos algoritmos

Os algoritmos escolhidos para a análise foram: (a) Regressão Logística (RL), (b) Árvore de Decisão Aumentada (AD), (c) Floresta de Decisão (FD), (d) Rede Neural (RN). Para todos os modelos foi utilizado a variante de duas classes, que visa uma classificação binária. Na “Tabela 2” são expostos os parâmetros utilizados:

Tabela 2. Parâmetros utilizados para a criação de modelos preditivos

Parâmetro	Regressão Linear	Árvore de Decisão	Floresta de Decisão	Rede Neural
Tolerância de Otimização	1E-07	x	x	x
Peso de Regularização L1	1	x	x	x
Peso de Regularização L2	1	x	x	x
Tamanho da Memória	20	x	x	x
Máximo número de folhas por árvore	x	20	x	x
Mínimo número de instâncias de treinamento para formar uma folha	x	2	x	x
Taxa de aprendizado	x	0,2	x	0,1
Número total de árvores construídas	x	100	x	x
Método de reamostragem	x	x	Encascamento	x
Número de árvores de decisão	x	x	8	x
Profundidade máxima da árvore de decisão	x	x	32	x
Número de divisões aleatórias por nós	x	x	128	x
Número mínimo de amostras pro nó folha	x	x	1	x
Especificação de camada oculta	x	x	x	Totalmente conectado
Número de nós ocultos	x	x	x	100
diâmetro da taxa de aprendizagem inicial	x	x	x	0,1
Momento	x	x	x	0
Tipo de normalizador	x	x	x	Min-Max

4. RESULTADOS E DISCUSSÕES

4.1 Apresentação e discussão geral de resultados

Ao finalizar a avaliação dos modelos dentro do Microsoft Azure, a ferramenta nos traz alguns parâmetros indicativos tais como a acurácia do modelo e o número de verdadeiros e falsos positivos e negativos que são importantes para a análise e comparação dos algoritmos. Esses parâmetros dependem do limite (Threshold) escolhido para a determinação de um evento como falha. Por exemplo, caso seja escolhido um limite de 0,5, casos classificados com 50% ou mais de chance de serem falhas serão classificados como tal. Visando simplificar a análise, os limites escolhidos foram os de: 0,3; 0,5; 0,75; 0,9. Os resultados estão expressos na matriz de confusão da Tab. 3:

Tabela 3. Resultados da avaliação de modelos preditivos.

Algoritmo	Limite (Threshold)	Verdadeiro positivos (VP)	Falsos Negativos (FN)	Falso Positivo (FP)	Verdadeiro Negativo (VN)	% Verdadeiro Positivo	% Verdadeiro Negativo	Acurácia
RL	0,3	39	0	43	2	1,000	0,044	0,496
RL	0,5	19	17	13	32	0,528	0,711	0,630
RL	0,75	9	27	0	45	0,250	1,000	0,667
RL	0,9	2	34	0	45	0,056	1,000	0,580
AD	0,3	36	0	7	38	1,000	0,844	0,914
AD	0,5	35	1	6	39	0,972	0,867	0,914
AD	0,75	35	1	4	41	0,972	0,911	0,938
AD	0,9	34	2	1	44	0,944	0,978	0,963
FD	0,3	35	1	7	38	0,972	0,844	0,901
FD	0,5	34	2	4	41	0,944	0,911	0,926
FD	0,75	27	9	1	44	0,750	0,978	0,877
FD	0,9	11	18	0	45	0,500	1,000	0,778
RN	0,3	36	0	12	33	1,000	0,733	0,852
RN	0,5	36	0	8	37	1,000	0,882	0,901
RN	0,75	32	4	4	41	0,889	0,911	0,901
RN	0,9	24	12	2	43	0,667	0,956	0,827

A acurácia dos modelos preditivos apresentados na Tabela 3 é a razão entre o total de classes retornadas corretamente pelo total de classes classificadas. Para duas classes, é definida por: VP + VN / (VP + FP + VN + FN).

Dentre os 4 modelos analisados, nota-se que o algoritmo de regressão linear foi aquele que apresentou os piores resultados em termos de acurácia, alcançando um valor máximo de 66,7% quando utilizado o limite de 0,75. Logo, em uma primeira observação podemos concluir que o algoritmo de regressão logística não consegue classificar e prever satisfatoriamente os eventos de falha. Outro fator que contribui para essa conclusão é a combinação entre as taxas de verdadeiro positivo e negativo. Para limites baixos, a taxa de verdadeiros positivos se aproxima de 100% e a taxa de verdadeiros negativos se aproxima de 0% rapidamente, enquanto que para limites elevados, o inverso é observado. Esse tipo de resultado é esperado de algoritmos que tendem a fazer previsões próximas à previsão aleatória.

Com exceção do algoritmo de regressão logística, os três algoritmos restantes apresentaram resultados satisfatórios com acurácia mínima de 90,1%, quando analisados os limites ótimos de cada um. Assim, para uma comparação assertiva e recomendação de um modelo preditivo para testes complementares, se torna necessária também uma análise do melhor limite de cada um deles visando um equilíbrio entre as taxas de verdadeiro positivo e negativo.

Como apresentado, a exatidão do algoritmo da rede neural chegou ao valor de 90,1%, quando utilizados os limites de 0,5 e 0,75. Para o limite de 0,5 a taxa de verdadeiros positivos foi de 100% e a taxa de verdadeiros negativos foi de 82,2%. Quando aumenta-se o limite para 0,75, a taxa de verdadeiros positivos cai para 88,9% e a taxa de verdadeiros negativos sobe para 91,1%. Em um ambiente real de manufatura, considera-se preferível uma parada preventiva da máquina visando evitar uma falha a uma parada obrigatória da linha devido a uma falha, visto que uma paralisação abrupta na linha de montagem gera um custo muito maior. Seguindo esta premissa, conclui-se que o limite ótimo para o algoritmo de rede neural é o de 0,5.

O modelo de floresta de decisão, por sua vez, obteve resultados mais satisfatórios. Para o limite de 0,5, uma porcentagem de verdadeiros positivos de 94,4% e uma porcentagem de verdadeiros negativos de 91,1% foi obtida.

Finalmente, após um estudo aprofundado de todos os parâmetros, chega-se à conclusão de que o algoritmo com a melhor performance foi a árvore de decisão aumentada. Por meio da Tabela 3, observa-se uma acurácia calculada superior aos outros modelos, obtendo-se um valor máximo de 96,3% quando utilizado o limite de 0,9, e um valor mínimo de 91,4% quando utilizado o limite de 0,5.

A baixa dispersão dos valores da acurácia calculada para os diferentes limites corrobora para a validação de uma maior confiabilidade do modelo preditivo para uma possível aplicação em um ambiente real da manufatura. Para o limite de 0,9, obteve-se os seguintes resultados:

- Taxa de verdadeiros positivos: 94,44%
- Taxa de verdadeiros negativos: 97,8%

Em números absolutos, apenas 2 dentre os 36 valores da amostra de eventos de falha não foram classificados e apenas um dentre os 45 valores da amostra de eventos sem falha foram classificados como potenciais falhas.

4.2 Comparação de Gráficos ROC

As curvas ROC (do inglês Receiver Operating Characteristic) também podem ser de grande valia para a análise do desempenho dos algoritmos. Nesse gráfico é plotada a taxa de verdadeiros positivos versus a taxa de falsos positivos, de modo que, quanto mais próxima essa curva se aproximar do canto superior esquerdo, melhor será o desempenho do

classificador (que está maximizando a taxa de VP, minimizando a taxa de FP). Para as curvas que estão próximas à diagonal do gráfico, o resultado dos classificadores tendem a fazer previsões próximas à aleatória. Na Figura 3 são apresentadas a curva ROC, para os algoritmos de regressão linear e árvore de decisão.

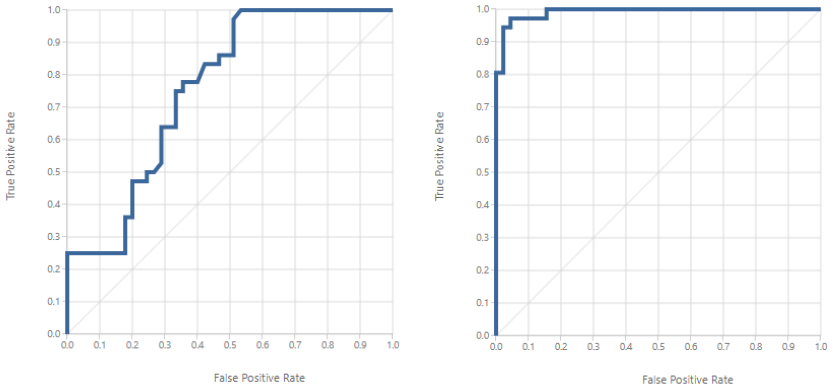

Figura 3. Gráficos ROC para os algoritmos de regressão linear (esquerda) e árvore de decisão (direita).

Quando comparados, os gráficos ROC acima apresentam uma diferença nítida no seu comportamento. Como a curva referente ao algoritmo de regressão linear se aproxima da diagonal do gráfico pode-se dizer que existe uma tendência à realização de previsões aleatórias. Essa mesma tendência a realizar previsões aleatórias foi observada através da análise dos resultados brutos.

O gráfico do algoritmo de árvore de decisão aumentada, por sua vez, foi aquele que mais apresentou um comportamento próximo ao ideal, reafirmando assim a sua melhor performance quando comparado aos outros algoritmos. Observando o gráfico à direita, na Figura 3, nota-se que a curva se aproximou significativamente do canto superior esquerdo do gráfico, o que caracteriza um bom desempenho do classificador.

4.3 Definição do modelo e próximos passos

Como discutido anteriormente o algoritmo com a melhor performance, e logo aquele recomendado para teste e provável aplicação futura é o modelo de árvore de decisão aumentada. Esse algoritmo produz, em geral, resultados melhores quando os recursos estiverem ligeiramente relacionados. Se os recursos tiverem um grande grau de entropia (ou seja, eles não estiverem relacionados), eles compartilharão pouca ou nenhuma informação mútua, e ordená-los em uma árvore não produzirá muita importância preditiva. Logo, pode-se concluir que tanto os parâmetros referentes à vibração quanto os valores referentes aos eventos de manutenção preventiva e planejada apresentam uma relação com as ocorrências de falha dentro da máquina.

No entanto, esse é um dos tipos de aprendizado que mais demandam memória e esforço da máquina para sua utilização. Para uma aplicação prática em meio industrial onde é necessário o processamento de um grande conjunto de dados, recomenda-se um estudo mais aprofundado do mesmo para a escolha do melhor algoritmo. Finalmente, recomenda-se também um estudo complementar e particular dos parâmetros inseridos na construção do algoritmo, tais como a taxa de aprendizado do modelo, visando aumentar a sua performance.

5. CONCLUSÕES

A identificação de falhas por parte de uma programação simples nos permitiu perceber o real valor do aprendizado de máquina, por meio do tratamento de dados e aplicação de algoritmos indicados para este tipo de análise. Pudemos contribuir com o desenvolvimento de um método prático que futuramente poderá servir para aplicações na empresa.

Hoje é possível encontrar diversos materiais de suporte e facilitadores para desenvolvimento de modelos de aprendizado de máquina, de tal maneira que não há necessidade de se ter um estudo profundo e especializado em programação e matemática. Os processos se tornaram mais intuitivos e, com uma compreensão razoável, é possível alterar parâmetros e chegar em um modelo próximo ao ideal por meio de experimentos e tentativas. Na visão dos autores, um bom resultado é atingido principalmente quando se entende o processo, pois juntamente com o conhecimento técnico e a capacidade de interpretação dos resultados é que o modelo preditivo será mais otimizado.

Espera-se que este trabalho possa servir de guia e incentivo para outros projetos, além de contribuição científica para o meio acadêmico e empresarial por meio da aplicação de algoritmos de aprendizado de máquina (Machine Learning) em predição de falhas na manutenção industrial.

6. REFERÊNCIAS

Baldissarelli, L. e Fabro, E., 2019. *Manutenção Preditiva na indústria 4.0*. Volume 7, 2º edição.
Berry, M. e Linoff, G., 2014. *Data Mining Techniques*.Wiley; Indianapolis, 2° edição.
Braga, A., Carvalho, A. e Ludemir, T., 2000. *Redes Neurais Artificiais: Teoria e Aplicações*; LTC; Rio de Janeiro, 2000.
Carvalho, T.P., Soares, F.A.A.M.N., Vitac, R., Francisco, R DA P., Bastoc, J.P. e Alcalá, S.G., 2019. "A systematic literature review of machine learning methods applied to predictive maintenance". In *Computers & Industrial Engineering 137, 2019.*
Guia de IA do Azure para soluções de manutenção preditiva. <https://docs.microsoft.com/pt-br/azure/machine-learning/team-data-science-process/predictive-maintenance-playbook#solution-templates-for-predictive-maintenance>.
Haykin, S., 1999. *Neural Networks: A comprehensive Foundation*. Pearson Prentice Hall, Ontario, 2° edição.
Nogare, D. e Zavaschi, T., 2016. *Análise Preditiva com Azure Machine Learning*. R. São Paulo, B2U Editora.
Two-Class Decision Forest. <https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/two-class-decision-forest?redirectedfrom=MSDN>.
Velloso, H.G. e Hora, H.R.M, 2019. "Classificação de falhas de um centro de usinagem: um estudo de caso utilizando árvore de decisão". In *Simpósio de pesquisa operacional e logística da marinha, 2019*, Rio de Janeiro, Brasil.
Wu, T., Sari, D.Y., Lin, B. e Chang, C., 2019. "Monitoring of punch failure in micro-piercing process based on vibratory signal and logistic regression". In *Int J Adv Manuf Technol, 2017.*
Zenisek, J., Holzinger, F. e Affenzeller, M., 2019. "Machine learning based concept drift detection for predictive maintenance". In *Computers & Industrial Engineering 137, 2019.*
Zhang, W., Yang, D e Wang, H., 2019. "Data-Driven Methods for Predictive Maintenance of Industrial Equipment: A Survey". In *IEEE SYSTEMS JOURNAL, VOL. 13, NO. 3, 2019.*

7. RESPONSABILIDADE PELAS INFORMAÇÕES

Os autores são os únicos responsáveis pelas informações incluídas neste trabalho.

MACHINE LEARNING ALGORITHMS APPLICATION TO CREATE A PREDICTIVE MODEL OF FAILURES IN A CNC MACHINE

Gianfranco Daroit Cavassin
Flavia Ayumi Ushiwata
Renato Fernandes Grottoli
Fernando Deschamps
Universidade Federal do Paraná, DEMEC, Av. Cel. Francisco H. dos Santos, 210 CEP 81531-990, Curitiba /PR.
gianfrancocavassin@ufpr.br
flavia.ayumi@ufpr.br
renato.grottoli@upfr.br
fernando.deschamps@ufpr.br

Abstract. *In the current manufacturing world, the role of maintenance has been receiving increasing attention while companies understand that maintenance, when well performed, can be a strategic factor to achieve corporate goals. With the present context of Industry 4.0 and the exponential amount of data extracted from production processes, the latest trends of maintenance leans towards a predictive approach, aiming at maintenance execution based on mathematical models. In this scenario, Machine Learning algorithms and methods have been emerging as a promising tool in predictive maintenance applications to prevent failures in equipment that make up production lines. This work aims to compare and further analyse four machine learning algorithms to identify failures in the machine. In the first stage, a brief presentation of the main concepts involved is made, and then, the methods and parameters used for the creation and evaluation of the models are discussed. Finally, each of the algorithms created are separately analysed and then compared in order to reach a possible solution for the prediction of failures applications in a real case of a CNC machine.*

Keywords*: Machine Learning, predictive maintenance, algorithm, flaw prediction*

RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.