

Detecção de expulsão em processos de soldagem a ponto por resistência: uma análise comparativa envolvendo métodos baseados em aprendizado de máquina

Bernardo W. Leal*, Roberto Z. Freire**

*Departamento de Engenharia Elétrica

** Programa de Pós-Graduação em Engenharia de Produção e Sistemas (PPGEPS),
Escola Politécnica (EP), Pontifícia Universidade Católica do Paraná (PUCPR), Rua Imaculada Conceição, 1155,
80215-901, Curitiba, Brasil
(e-mails:bernardo.leal@pucpr.edu.br, roberto.freire@pucpr.br).

Abstract: The use of predictive maintenance in the industry helps to reduce production costs, as it decreases unforeseen events when repairing machinery during idle periods. Several machine learning methods have been employed for failure detection during welding procedures, and in this work, Decision Trees, k -Nearest Neighbor, Naive Bayes, Support Vector Machines, and Adaptive Boosting algorithms were evaluated to detect failures associated to the expulsion at resistance spot welding procedures in the automotive industry. Therefore, the objective of this work was to analyze the performance of these algorithms to detect the expulsion in welding using data without labels obtained from a proprietary controller. Results showed the accuracy of each method concerning the selected database, indicating that machine learning techniques can be adapted for welding failure detection. Finally, the k -Nearest Neighbor method presented the best results among all techniques evaluated.

Resumo: A utilização de manutenção preditiva na indústria auxilia na diminuição de custos de produção, pois reduz imprevistos ao reparar equipamentos durante períodos ociosos de funcionamento. Diversos métodos de aprendizado de máquina têm sido empregados para detecção de falhas em processos de soldagem. Neste estudo, os algoritmos Árvores de Decisão, k -ésimo Vizinho mais Próximo, Naive Bayes, Máquina de Vetores de Suporte e *Boosting* Adaptativo foram avaliados levando em conta a detecção de falhas no processo de soldagem a ponto por resistência na indústria automotiva. Neste sentido, o objetivo deste trabalho foi analisar o desempenho desses algoritmos na detecção de expulsão em soldas utilizando dados, não rotulados, provenientes de um controlador proprietário. Os resultados obtidos apresentam a acurácia de cada modelo levando em conta o conjunto de dados selecionado, indicando que técnicas de aprendizado de máquina podem ser utilizadas para a detecção de falhas no processo de soldagem. Por fim, o método k -ésimo Vizinho mais Próximo se mostrou como o método mais preciso avaliado neste estudo.

Keywords: resistance spot welding; predictive maintenance; machine learning; classification; k -Nearest Neighbor; Decision Trees; Naive Bayes; Support Vector Machines; Adaptive Boosting.

Palavras-chaves: solda ponto por resistência; manutenção preditiva; aprendizado de máquina; classificação; Árvores de Decisão; k -ésimo Vizinho mais Próximo; Naive Bayes; Máquina de Vetores de Suporte; *Boosting* adaptativo.

1. INTRODUÇÃO

A manufatura impacta de forma significativa na economia e no progresso da sociedade. Devido a sua importância, o nível de desenvolvimento da indústria acaba sendo um dos indicadores considerados para mensurar o grau de desenvolvimento de um país. Neste contexto, um termo que vem sendo aceito no ambiente acadêmico é a indústria 4.0. Iniciativas para a modernização da indústria por meio do acesso à informação proveniente do chão de fábrica, as quais atingem os níveis mais altos da área de negócios, estão em destaque tanto no meio acadêmico quanto no meio industrial (Oztemel e Gursev, 2020).

Enquanto as universidades têm por objetivo investigar e definir o conceito de indústria 4.0, além de desenvolver

sistemas, modelos de negócio e metodologias que integrem todas as etapas da cadeia produtiva, a indústria foca nas mudanças que o conceito de indústria 4.0 pode trazer fazendo uso de sistemas inteligentes, em especial os baseados em inteligência computacional para identificar novos mercados e aumentar a competitividade por meio de novas tecnologias (Beier et al., 2020).

Um dos problemas enfrentados pela indústria moderna é a capacidade de realizar manutenção preditiva (COSTA et al., 2019), um problema que pode ser abordado por técnicas de inteligência computacional, em especial o aprendizado de máquina (do inglês, *Machine Learning* – ML). Os métodos de aprendizado de máquina vêm ganhando cada vez mais visibilidade na indústria, já que existe um aumento significativo na aquisição de dados relativos a processos no

contexto da indústria 4.0. Esses dados podem tornar o processo de manutenção preditiva mais fácil e eficiente quando comparado aos métodos de manutenção preventiva e manutenção na falha (Susto et al., 2015).

Um dos grandes problemas enfrentados com os dados é que uma boa parte deles pode ser proprietário, logo, o seu tratamento torna-se difícil sem a ajuda do fabricante do produto do qual é feito o processo de aquisição. Isso ocorre porque esses dados possuem uma formatação específica, sem rótulos intuitivos, ou até mesmo estão em forma criptografada. Uma maneira de abstrair a complexidade dos dados proprietários é por meio do uso de métodos inteligentes, capazes de interpretar e abstrair esta complexidade.

Preparar dados para serem utilizados por métodos de aprendizado de máquina é, geralmente, uma tarefa trabalhosa, pois estes dados necessitam de um tratamento prévio, como análises estatísticas. A análise de correlação ajuda a decidir quais dados devem ser utilizados no processo, e muitas vezes pode substituir os dados rotulados, estes definidos como os dados cuja resposta de destino já é conhecida (Shi et al., 2019).

No que diz respeito ao processo de soldagem por resistência, que está presente em diversos ramos da indústria eletromecânica, já é de conhecimento geral de especialistas no ramo a importância de cada parâmetro associado ao processo, mas nem sempre é possível ter acesso a todos os dados do controlador de solda. Então, é necessário fazer uma análise dos dados disponíveis para identificar a qualidade da solda, que impacta diretamente na qualidade do produto. Na sequência, apresentam-se alguns trabalhos envolvendo o processo de detecção de falhas em processos de soldagem, de forma a enfatizar a relevância do tema abordado neste artigo.

Martín et al. (2014) aplicaram algoritmos de aprendizado de máquina baseados em árvore, para processar os dados adquiridos por meio dos métodos de análise ultrassônica da solda. Os resultados da análise para todos esses métodos ultrapassaram 93% de precisão. O método de aprendizado de máquina chamado Floresta Aleatória (do inglês, *Random Forest* – RF) foi utilizado para situações específicas, alcançando precisão de 99% nas classificações de solda.

Kim, Park e Sohmshtetty (2017) mostraram que existe uma inconsistência nos dados do processo de solda a ponto, e que as vezes a análise pelo algoritmo de aprendizado de máquina *k*-ésimo Vizinho mais Próximo (do inglês *k-Nearest Neighbor* – *kNN*) se torna insuficiente para o caso, o que levou os autores a desenvolverem um método estatístico para auxiliar na precisão do algoritmo, chamado de Média do Erro Aceitável (MACE), o qual aumentou a taxa de acerto do algoritmo considerando bases de dados inconsistentes.

Ahmed e Kim (2017) também utilizaram o algoritmo de Árvores de decisão para prever a largura dos pontos de solda por resistência. Ao analisar a corrente de solda, tempo de solda e espessura do material, a eficácia do método foi comprovada através do Erro Médio Quadrático (do inglês, *Mean Squared Error* – MSE), mostrando pouca variação nos métodos de treino utilizados.

Xing et al. (2018) conseguiram aplicar o método de aprendizado de máquina RF no sinal de resistência dinâmica durante o processo de soldagem para detectar soldas boas, soldas frias e expulsão, analisando a correlação entre os dados coletados. No estudo, os autores alcançaram precisões entre 93,6% até 98,8%, dependendo dos dados utilizados no método.

Já no trabalho apresentado por Zhang et al. (2019) considerou-se a soldagem a arco para ligas de alumínio. A análise da qualidade da soldagem foi feita em tempo real, também utilizando o algoritmo RF. Neste caso, uma câmera e um microfone foram empregados no processo para ajudar na identificação dos casos devido à importância dos sinais visuais e auditivos do processo de solda a arco. Como resultados, os autores atingiram uma precisão de 97,75% na classificação das soldas.

No estudo proposto por Zhao et al. (2020), o desempenho de um modelo de regressão de segundo grau e de uma rede neural artificial treinada através do algoritmo de retropropagação do erro foram comparadas em relação ao monitoramento do tamanho da solda pelo sinal de corrente, força do eletrodo, tempo de solda e tempo de sustentação. O estudo comprovou que o método de rede neural artificial, quando comparado com a abordagem clássica por regressão, é mais preciso na classificação do caso em questão.

No contexto dos trabalhos apresentados acima, envolvendo soldagem em linhas de produção, este trabalho tem por objetivo utilizar os dados não rotulados proprietários de um controlador de solda em algoritmos de aprendizado de máquina, onde o intuito é identificar e reportar casos de solda no qual ocorreram expulsão, que é um fenômeno conhecido por alterar a resistência da solda, ocasionado pelo aquecimento excessivo do material soldado devido a falha de algum dos parâmetros durante o processo, resultando no espirro de material derretido. O objetivo é aprimorar o controle de qualidade na soldagem a ponto das máquinas, pois esse fenômeno impacta diretamente na qualidade do produto.

Após a revisão de literatura, que reuniu conhecimentos de situações similares e de técnicas utilizadas para identificação de falhas nos processos de soldagem, foram definidos os métodos que seriam utilizados ao longo deste estudo. Os algoritmos selecionados foram: Árvores de Decisão (do inglês, *Decision Trees* – DT) (Quinlan, 1986), Naive Bayes (NB) (Kononenko, 1991), Máquinas de Vetores de Suporte (do inglês, *Support Vector Machines* – SVM) (Cortes e Vapnik, 1995), *k*-ésimo Vizinho mais Próximo (kNN) (Cover e Hart, 1967), e o método *Boosting* Adaptativo (do inglês, *Adaptive Boosting* – AdaBoost) (Freund e Schapire, 1997). Logo após, foi realizada a coleta de dados para os testes dos algoritmos de aprendizado de máquina e, em seguida, foi feita a adaptação dos métodos ao problema e a análise dos resultados obtidos através da análise comparativa de desempenho.

A próxima seção deste artigo descreve o problema geral abordado no âmbito da pesquisa, definindo as situações enfrentadas pelo processo. Na seção 3 apresenta-se uma breve descrição dos métodos utilizados para que haja uma ambientação do leitor com os algoritmos utilizados na pesquisa. Na sequência, a seção 4 define um estudo de caso

utilizado nesta pesquisa vinculado a um processo de soldagem na indústria automotiva, exemplificando equipamentos, aquisição e pré-processamento de dados. Os resultados dos algoritmos bem como a análise comparativa são apresentados na seção 5. Por fim, a seção 6 descreve as conclusões do trabalho, uma breve discussão da precisão obtida pelos métodos e ainda inclui sugestões para trabalhos futuros.

2. DETECÇÃO DE FALHAS NA SOLDAGEM EM LINHAS DE PRODUÇÃO

As máquinas de solda a ponto por resistência possuem um papel fundamental na indústria automobilística, onde manter o padrão de qualidade dos pontos de solda pode ser um desafio devido a diversos fatores, alguns inclusive fora das configurações da máquina, como por exemplo o posicionamento das peças a serem soldadas. O processo de soldagem é dado pela pinça de solda, que prensa o material, e injeta uma alta corrente para elevar a temperatura na área de contato entre as placas, fazendo com que o material derreta, fundindo as duas peças. Todavia, a corrente de soldagem não é fixa, pois há uma relação desse parâmetro com as resistências térmicas dos materiais que, por sua vez, variam com a temperatura. Além disso, a expansão térmica pode afetar a área de contato entre as placas, quando estas estiverem mal posicionadas, ocasionando falhas no processo de soldagem. Outro fator preponderante que altera a qualidade do produto é o desgaste da máquina, porém nesse caso, existem vários métodos de prevenção, sejam eles preditivos ou preventivos (Ru-xiong, 2012).

A aquisição dos dados da máquina utilizada neste estudo foi realizada por meio de um controlador de solda que, dependendo do modelo ou da licença, fornece informações suplementares sobre o processo. A partir dessas informações, procurou-se investigar os padrões presentes nos conjuntos de dados provenientes das máquinas que possuíam o controlador, para que pudessem ser aplicados nas máquinas que não possuíam a mesma extensão de monitoramento. Desta forma, o foco foi desenvolver uma abordagem para classificar erros no processo de soldagem a ponto por resistência, sem a necessidade do equipamento proprietário.

A base de dados utilizada na pesquisa foi retirada do controlador que possuía as informações extras citadas anteriormente, junto com outra base de dados de controladores que não possuíam estas informações. A separação das informações do controlador inicial foi realizada em pontos de solda em que ocorreram expulsão e em pontos em que não ocorreram, com o intuito de detectar esse caso em outros controladores, para assim prever possíveis falhas de soldagem.

3. TÉCNICAS DE APRENDIZADO DE MÁQUINA: UM PROBLEMA DE CLASSIFICAÇÃO

Nesta seção foram descritos os algoritmos de aprendizado de máquina selecionados para serem testados no contexto desta pesquisa. A seguir, apresenta-se uma breve revisão teórica sobre cada técnica utilizada.

3.1 Árvores de Decisão (DT)

O algoritmo de aprendizado de máquina de Árvores de decisão (DT) foi desenvolvido por Quinlan (1986), e pode ser definido como uma estrutura recursiva, na qual é feita uma classificação em sequência, até chegar nas folhas da árvore que, neste caso, são as classes. Cada nó que antecede uma folha representa um teste que possui atributos de classificação de caso com uma decisão para cada resultado obtido e, assim, é feita a escolha no algoritmo, da raiz até a folha. O método analisa o conjunto de dados (*dataset*) considerando dados rotulados para aplicar os atributos de classificação nos nós da árvore, para que seja feito um caminho até a classificação. Porém, o algoritmo pode não conseguir prever casos raros que não tenham sido notificados no *dataset* de referência. A raiz da árvore dá origem a todas as classes possíveis e os nós que foram criados a partir dela darão origem a várias outras classes se necessário, até chegar na folha, que é o resultado da classificação. A Fig. 01 representa um exemplo de criação das árvores de decisão do algoritmo.

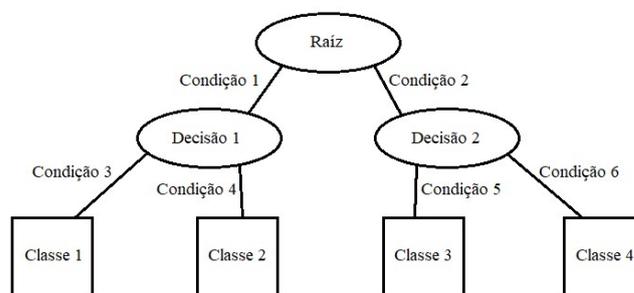


Fig. 1: Exemplo de uma DT (adaptado de Safavian e Landgrebe, 1991).

Para classificação de dados nunca vistos pela árvore, o algoritmo precisa capturar a relação entre as classes e seus valores de atributos para que seja possível fazer uma escolha coerente com os dados de treinamento, esse requisito aumenta significativamente a complexidade da árvore montada. Um novo nó de decisão é feito para cada característica nova de atributo encontrada.

3.2 *k*-Nearest Neighbor (*k*NN)

No método proposto por Cover e Hart (1967), a classificação é dada pelo vizinho mais próximo do ponto analisado no espaço, ou dos seus *k*-vizinhos. Quando um ponto está próximo de mais de um tipo de vizinho, esses *k*-vizinhos determinarão a classificação do ponto por meio de uma votação, que conta o número de tipos de vizinhos próximos e elege o que tiver mais pontos.

Na consideração do valor de *k*, deve-se levar em conta que um *k* pequeno pode deixar o resultado ruidoso, e assim, atrapalhar na classificação de novos dados. Já valores de *k* grande, podem suavizar esse ruído, porém, esse *k* não pode ser maior do que o necessário, para que um ponto não seja classificado erroneamente, isso por ter uma proporção menor de dados que as outras classificações. Um exemplo do funcionamento do algoritmo pode ser visto na Fig. 2.

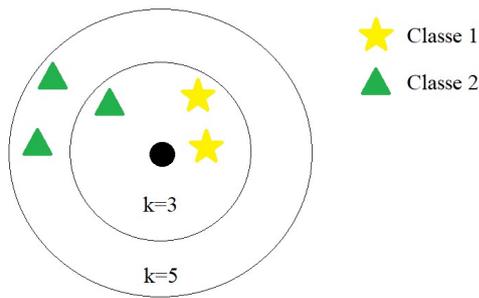


Fig. 2: Exemplo do funcionamento do algoritmo kNN (adaptado de Huang, 2014).

Neste caso, quando o algoritmo funcionar com $k = 3$, a classificação do novo ponto será a classe 1, e se $k = 5$, será classe 2.

Quando uma votação termina em empate, o algoritmo tem outras saídas para determinar essa classificação. O método pode definir o tipo do dado por decisão aleatória, como também pode definir novos valores de k para que ocorra o desempate.

3.3 Naive Bayes (NB)

O algoritmo NB foi proposto por Kononenko (1991), este método consiste em montar histogramas com as características de cada classe que compõe o *dataset*, contabilizando todos as ocorrências de valores nos campos de cada característica.

A classificação de um dado sem rótulo é dada por probabilidades, as quais são calculadas considerando as ocorrências de todas as características pertencentes a uma determinada classe. Quando um dado novo chega, essas características são contabilizadas e, então, calcula-se o valor da probabilidade de cada uma dessas características aparecer em cada classe. A classe que tiver a maior probabilidade é a classificação que o dado irá receber pelo método. Uma exemplificação do funcionamento do método é apresentada na Fig. 3.

É importante notar que, como o método trata de cálculos de probabilidade, existem certas técnicas que são utilizadas no algoritmo para evitar que algumas combinações específicas de características gerem ruído nas classificações. Algumas dessas técnicas podem ser encontradas em Xu (2018).

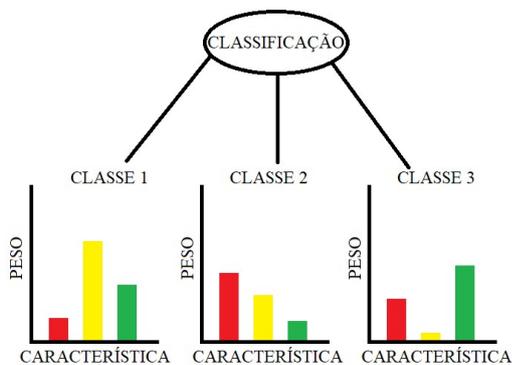


Fig. 3: Exemplo de funcionamento do método NB (adaptado de Xu, 2018).

3.4 Boosting Adaptativo (AdaBoost)

A ideia por trás do algoritmo proposto no trabalho de Freund e Schapire (1997) é utilizar pequenas árvores de decisão para votação no processo de classificação, considerando pesos diferentes nos votos, levando em consideração os erros anteriores da árvore.

No início da análise, cada amostra tem o mesmo peso, então, não existem muitos critérios na iteração do método. O novo peso de cada variável é calculado com o número de erros que o classificador teve na escolha da classe, o que tiver o menor número de erros terá o maior peso no voto geral.

O peso de escolha desses serve para colocar um novo peso nas amostras, e essas amostras servem para criar um espaço amostral atualizado de mesmo tamanho, considerando amostras aleatórias do espaço anterior. As amostras que tiverem mais peso aparecem mais vezes nesse espaço, criando blocos, que perdem muito mais peso se forem classificados erroneamente. Um exemplo do funcionamento do AdaBoost e como ele utiliza um conjunto de classificadores mais simples para melhorar a qualidade de classificação é apresentado na Fig. 4.

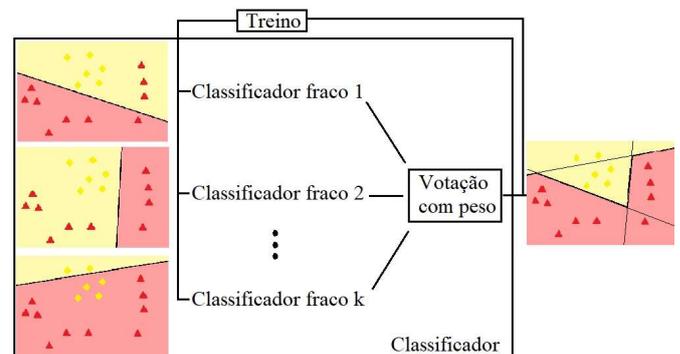


Fig. 4: Exemplo de classificação através do algoritmo AdaBoost (adaptado de Wang et al., 2015).

A classificação final é dada pela soma do peso das classificações das árvores criadas, a classificação que tiver a maior somatória ganha como sendo a decisão final do algoritmo.

3.5 Máquina de Vetores de Suporte (SVM)

O método SVM foi desenvolvido por Cortes e Vapnik (1995) e, como o próprio nome supõe, mapeia um vetor de entrada em um alto espaço dimensional Z por algum mapeamento não linear definido, construindo uma região neste espaço, que pode ser descrita como uma superfície de decisão linear. Neste sentido esse espaço possui uma propriedade que garante a habilidade de alta generalização do método.

Dois fatores devem ser controlados quando se fala da habilidade de generalização de uma SVM: a taxa de erro do modelo de treinamento e a capacidade da máquina de aprendizado. Quanto menor a capacidade da máquina, menor é o intervalo de confiança, nesse caso, o valor de frequência de erro é maior. Porém, essa troca pode ser controlada alterando parâmetros do algoritmo.

Em termos de classificação, o algoritmo utilizado neste estudo considera um limiar que foi calculado através do *kernel* polinomial para criar o hiperplano para fazer a seleção dos dados, observando no espaço os pontos que estão espalhados próximos ao limiar, e então aprender as classificações. Neste sentido, o algoritmo possui proteção em relação a classificações erradas quando o ponto está muito perto de outros com rotulagem diferente da sua. Um exemplo de classificação feita pelos vetores de suporte pode ser visualizada na Fig. 5.

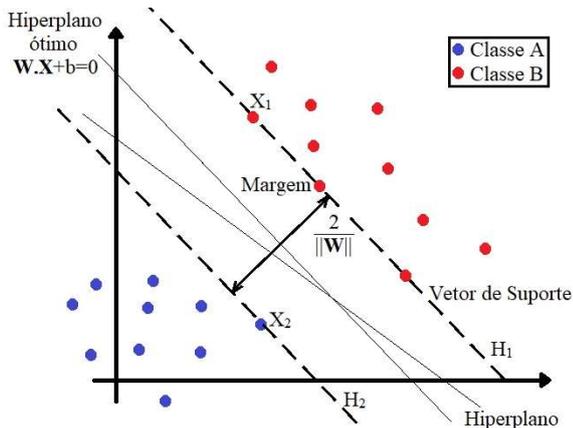


Fig. 5: Exemplo de classificação usando a SVM (adaptado de García-Gonzalo et al., 2016).

4. ESTUDO DE CASO NA INDÚSTRIA AUTOMOTIVA

Nesta seção estão descritas as atividades realizadas com a base de dados que foi coletada do controlador de solda de uma das linhas de produção da fábrica.

4.1 Equipamentos e Aquisição de Dados

A base de dados foi retirada do controlador de solda que possuía a distinção de pontos de solda com e sem expulsão, para que fosse possível a classificação dos casos. Esses dados são referentes a um total de 18 máquinas de solda da linha de produção que são monitorados pelo controlador. Os dados ficam armazenados no disco rígido do computador no qual está instalado o programa do controlador de solda, e o arquivo está em formato atributo-valor, utilizando texto simples.

A base de dados possui um pouco mais de 100.000 pontos de solda e 115 colunas com informações sobre o processo de solda a ponto por resistência, estes referentes a 7 horas de produção. Dessas 115 colunas, apenas 13 colunas foram consideradas relevantes para a análise, e 3 dessas são apenas colunas de identificação do processo. Não se tem nenhuma informação do que os dados armazenados nessas colunas representam, por serem de um software proprietário.

4.2 Pré-processamento de Dados

Para o processamento dos dados, foi necessário converter o formato dos dados, melhorando a visualização das informações, além de possibilitar a integração com outros programas computacionais.

Dos 100.000 pontos de solda, apenas 7.300 foram classificados como pontos de solda com expulsão, então foram separados 14.600, 50% desses dados são solda com expulsão, os outros 50% sem expulsão, escolhidos aleatoriamente entre os dados das 18 máquinas. O objetivo desta seleção foi realizar o balanceamento das classes.

Os dados de todas as máquinas foram misturados e separados igualmente em 5 *datasets*, para facilitar na análise dos algoritmos de aprendizado de máquina, sendo utilizados em um método de validação cruzada para chegar em uma conclusão fundamentada sobre o desempenho em termos de precisão dos métodos utilizados, melhorando assim sua confiabilidade. Mais informações sobre esse tópico serão incluídas na seção apresentada a seguir.

Nenhum dos dados que seriam considerados como ruído foram removidos do banco de dados, com o objetivo de analisar a precisão dos algoritmos com as informações retiradas diretamente do controlador de solda da linha de produção que não possuem nenhum tipo de tratamento prévio.

5. RESULTADOS

Nesta seção serão discutidos os resultados de precisão dos algoritmos de aprendizado de máquina selecionados, bem como os tempos de treinamento e de classificação.

5.1 Parâmetros dos Modelos e Configuração do Experimento

Das 13 séries de dados que poderiam ser relevantes para o estudo, apenas 5 demonstraram ser úteis na utilização dos modelos de aprendizado de máquina. Os parâmetros utilizados nos algoritmos foram escolhidos levando em conta as informações e características analisadas na base de dados, escolhendo o melhor caminho para facilitar e agilizar o processo de obtenção dos resultados.

Todos os métodos e implementações foram adaptados considerando o ambiente MATLAB.

5.2 Treinamento e Validação

Para realizar o treinamento e a validação dos algoritmos de aprendizado de máquina selecionados, optou-se pela validação cruzada (do inglês, *K-fold cross validation*). Considerou-se $K = 5$. Este método consiste em separar os dados igualmente em K *datasets* menores. Neste caso, com o valor de K escolhido, 80% desses *datasets* são utilizados para o treinamento do método, e os outros 20% que sobraram servem para testar o algoritmo, para que não sejam usados dados já vistos em treinamento, evitando que o algoritmo forneça informações falsas de sua precisão na classificação.

Nas tabelas apresentadas a seguir é possível visualizar os resultados de validação obtidos levando em consideração os resultados do melhor *dataset* para cada algoritmo e o resultado da validação cruzada 5-Fold. Os resultados apresentados em negrito representam os melhores resultados entre todos os métodos levando em conta o custo computacional.

Tabela 1. Resultados de validação do algoritmo DT.

Parâmetros	Melhor Dataset	5-Fold
Precisão (%)	86,38	85,43
Classificações Erradas	407	427
Velocidade de Classificação (obs/s)	2,2E6	2,04E6
Tempo de Treinamento (s)	0,97	1,04

O algoritmo DT, com número de divisões máxima de 100, alcançou a maior velocidade de classificação e menor tempo de treinamento, com a segunda melhor precisão dos métodos.

Tabela 2. Resultados de validação do algoritmo kNN.

Parâmetros	Melhor Dataset	5-Fold
Precisão (%)	91,43	90,70
Classificações Erradas	251	273
Velocidade de Classificação (obs/s)	310.000	264.000
Tempo de Treinamento (s)	1,30	1,52

Os resultados do método kNN, com $k = 1$ e métrica de distância euclidiana, foram os melhores entre os métodos selecionados, mesmo utilizando o *dataset* sem remoção dos ruídos.

Tabela 3. Resultados de validação do algoritmo NB.

Parâmetros	Melhor Dataset	5-Fold
Precisão (%)	78,13	77,79
Classificações Erradas	640	680
Velocidade de Classificação (obs/s)	1,9E3	1,86E3
Tempo de Treinamento (s)	19,21	19,56

O método NB, com kernel Gaussiano, teve um desempenho inferior em velocidade de classificação quando comparado a todos os outros métodos, principalmente devido ao modelo do *dataset*, que não favoreceu o método estatístico.

Tabela 4. Resultados de validação do algoritmo AdaBoost.

Parâmetros	Melhor Dataset	5-Fold
Precisão (%)	84,38	83,60
Classificações Erradas	457	480
Velocidade de Classificação (obs/s)	120E3	116,4E3
Tempo de Treinamento (s)	2,57	2,26

O método AdaBoost, com 20 árvores e 30 estimadores teve um resultado razoável, mas com velocidade de classificação limitada comparada ao método DT.

Tabela 5. Resultados de validação do algoritmo SVM.

Parâmetros	Melhor Dataset	5-Fold
Precisão (%)	76,56	75,33
Classificações Erradas	686	722
Velocidade de Classificação (obs/s)	39E3	40,2E3
Tempo de Treinamento (s)	205,19	116,86

O método SVM, com Kernel linear, foi o mais lento no treinamento, e apresentou problemas na sua precisão devido ao *dataset* escolhido.

6. CONCLUSÕES

Este trabalho aplicou 5 métodos de aprendizado de máquina: Árvores de Decisão, Naive Bayes, Máquinas de Vetores de Suporte k -Vizinhos Mais Próximos e o método *Boosting* Adaptativo, todos voltados à detecção de expulsão em pontos de solda por resistência, utilizando um banco de dados que foi retirado da linha de produção da indústria automotiva. Estes dados passaram por uma etapa de pré-processamento para que pudessem ser utilizados de maneira correta na geração dos resultados fornecidos pelas técnicas de aprendizado de máquina avaliadas neste trabalho. Um estudo de caso foi feito para a seleção dos algoritmos de aprendizado de máquina, visando definir quais métodos trariam maior desempenho.

Como mostrado anteriormente, inúmeras pesquisas já foram desenvolvidas estudando os métodos de aprendizado de máquina para o processo de soldagem a ponto por resistência. A diferença do estudo atual para as outras pesquisas é que nesta situação específica da indústria, não se tinha muita informação ou rotulação nos dados que estavam sendo utilizados nos métodos, e também não foi feito nenhum tipo de remoção de ruído desses dados, apenas o estudo de correlação entre as entradas de dados foram analisadas, o que tornava necessário um estudo específico do caso.

No estudo realizado, o método de k -ésimo Vizinho mais Próximo trouxe o melhor resultado entre os métodos, com um parâmetro simples de apenas 1 vizinho próximo, com precisão de 90,70% na classificação de pontos de solda com expulsão, seu tempo de treinamento foi rápido e sua velocidade de classificação considerada interessante para aplicações em tempo real, que devem ser o foco dos trabalhos futuros associados a esta pesquisa. Com a utilização do método de validação cruzada, foi possível analisar a consistência dos métodos, mostrando pouca variância na precisão nos testes realizados.

Um dos fatores que impactou na precisão dos algoritmos foi a inconsistência do banco de dados, já que o processo de solda a ponto por resistência depende de muitos parâmetros considerados dinâmicos. Os algoritmos também não possuíam uma unidade de pré-processamento que considerasse uma proteção contra ruídos dos conjuntos de dados, o que interferiu no resultado, minimizando os acertos dos métodos na classificação.

Em relação aos trabalhos futuros, sugere-se avaliar diferentes hiperparâmetros dos métodos tentando aumentar a precisão no processo de classificação. Além disso, sugere-se ainda a otimização dos métodos de aprendizado de máquina por meio de outras técnicas, como a otimização metaheurística, para que essas informações possam ser utilizadas com maior confiabilidade em projetos de manutenção preditiva na indústria automotiva, diminuindo o tempo ocioso da linha de produção.

AGRADECIMENTOS

Os autores agradecem a Fundação Araucária do Estado do Paraná (Chamada Pública 06/2019) e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) (Processo: 304783/2017-0), pelo financiamento desta pesquisa.

REFERÊNCIAS

- Ahmed, F. and Kim, K.-Y. (2017), “Data-driven Weld Nugget Width Prediction with Decision Tree Algorithm”, *Procedia Manufacturing*, The Author(s), Vol. 10, pp. 1009–1019.
- Beier, G., Ullrich, A., Niehoff, S., Reißig, M. and Habich, M. (2020), “Industry 4.0: How it is defined from a sociotechnical perspective and how much sustainability it includes – A literature review”, *Journal of Cleaner Production*, Vol. 259, p. 120856.
- Cortes, C. and Vapnik, V. (1995), “Support-Vector Networks”, *Machine Learning*, Vol. 20, pp. 273–297.
- COSTA, G.K., Kagami, R.M., Diogo, R., Zanetti Freire, R. and SANTOS, N. (2019), “Comparação de modelos de previsão voltados à manutenção preditiva na indústria automobilística a partir de dados de inspeção de qualidade”, *Anais Do 14º Simpósio Brasileiro de Automação Inteligente*, Galoa, available at: <https://doi.org/10.17648/sbai-2019-111404>.
- Cover, T.M. and Hart, P.E. (1967), “Nearest Neighbor Pattern Classification”, *IEEE Transactions on Information Theory*, Springer US, Boston, MA, Vol. 13 No. 1, pp. 21–27.
- Freund, Y. and Schapire, R.E. (1997), “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting”, *Journal of Computer and System Sciences*, Vol. 55 No. 1, pp. 119–139.
- García-Gonzalo, E., Fernández-Muñiz, Z., García Nieto, P., Bernardo Sánchez, A. and Menéndez Fernández, M. (2016), “Hard-Rock Stability Analysis for Span Design in Entry-Type Excavations with Learning Classifiers”, *Materials*, Vol. 9 No. 7, p. 531.
- Huang, Y.-K. (2014), “Processing KNN Queries in Grid-Based Sensor Networks”, *Algorithms*, Vol. 7 No. 4, pp. 582–596.
- Kim, K.-Y., Park, J. and Sohmshtetty, R. (2017), “Prediction measurement with mean acceptable error for proper inconsistency in noisy weldability prediction data”, *Robotics and Computer-Integrated Manufacturing*, Elsevier, Vol. 43, pp. 18–29.
- Kononenko, I. (1991), “Semi-naive bayesian classifier”, *Machine Learning — EWSL-91*, Vol. 482 LNAI, Springer-Verlag, Berlin/Heidelberg, pp. 206–219.
- Martín, Ó., Pereda, M., Santos, J.I. and Galán, J.M. (2014), “Assessment of resistance spot welding quality based on ultrasonic testing and tree-based techniques”, *Journal of Materials Processing Technology*, Elsevier B.V., Vol. 214 No. 11, pp. 2478–2487.
- Oztemel, E. and Gursev, S. (2020), “Literature review of Industry 4.0 and related technologies”, *Journal of Intelligent Manufacturing*, Springer US, Vol. 31 No. 1, pp. 127–182.
- Quinlan, J.R. (1986), “Induction of decision trees”, *Machine Learning*, Vol. 1 No. 1, pp. 81–106.
- Ru-xiong, L.I. (2012), “Quality monitoring of resistance spot welding based on process parameters”, *Energy Procedia*, Vol. 14, pp. 925–930.
- Safavian, S.R. and Landgrebe, D. (1991), “A survey of decision tree classifier methodology”, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 21 No. 3, pp. 660–674.
- Shi, M., Zhang, L., Sun, W. and Song, X. (2019), “A fuzzy c-means algorithm guided by attribute correlations and its application in the big data analysis of tunnel boring machine”, *Knowledge-Based Systems*, Vol. 182, p. 104859.
- Susto, G.A., Schirru, A., Pampuri, S., McLoone, S. and Beghi, A. (2015), “Machine Learning for Predictive Maintenance: A Multiple Classifier Approach”, *IEEE Transactions on Industrial Informatics*, Vol. 11 No. 3, pp. 812–820.
- Wang, Z., Zhang, J. and Verma, N. (2015), “Realizing Low-Energy Classification Systems by Implementing Matrix Multiplication Directly Within an ADC”, *IEEE Transactions on Biomedical Circuits and Systems*, Vol. 9 No. 6, pp. 1–1.
- Xing, B., Xiao, Y., Qin, Q.H. and Cui, H. (2018), “Quality assessment of resistance spot welding process based on dynamic resistance signal and random forest based”, *The International Journal of Advanced Manufacturing Technology*, The International Journal of Advanced Manufacturing Technology, Vol. 94 No. 1–4, pp. 327–339.
- Xu, S. (2018), “Bayesian Naïve Bayes classifiers to text classification”, *Journal of Information Science*, Vol. 44 No. 1, pp. 48–59.
- Zhang, Z., Yang, Z., Ren, W. and Wen, G. (2019), “Random forest-based real-time defect detection of Al alloy in robotic arc welding using optical spectrum”, *Journal of Manufacturing Processes*, Elsevier, Vol. 42 No. April, pp. 51–59.
- Zhao, D., Wang, Y., Liang, D. and Ivanov, M. (2020), “Performances of regression model and artificial neural network in monitoring welding quality based on power signal”, *Journal of Materials Research and Technology*, The Authors, Vol. 9 No. 2, pp. 1231–1240.