

Intelligent Hybrid Method Based on 1D-CNN and SVM Applied to Gear Fault Diagnosis

Vinícius Serra Vianna, v178334@dac.unicamp.br¹

Milton Dias Junior, milton@fem.unicamp.br¹

¹Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, Mendeleiev st, 200, Cidade Universitária “Zeferino Vaz”, 13098-360 Campinas, São Paulo, Brazil

Resumo. As características não lineares e não estacionárias de sistemas reais são frequentemente difíceis de serem extraídas. O presente trabalho concentra-se no desenvolvimento de uma Rede Neural Convolutiva (CNN) para obter atributos diretamente dos sinais de vibração originais de uma caixa de engrenagens com diferentes condições do pinhão. Os dados experimentais são utilizados para demonstrar a eficiência do método apresentado. Máquinas de Vetores de Suporte (SVM) são utilizadas para classificar os atributos extraídos com 1D-CNN. Os resultados obtidos mostram que os atributos extraídos neste método apresentam excelente qualidade para classificação de falhas sem qualquer seleção de atributos adicional.

Palavras chave: Caixa de Engrenagens, 1D-CNN, SVM, Detecção de falha

Abstract. The features of nonlinearity and non-stationarity in real systems are often difficult to be extracted. This paper focuses on developing a Convolutional Neural Network (CNN) to obtain features directly from the original vibration signals of a gearbox with different pinion conditions. Experimental data is used to show the efficiency of the presented method. Support Vector Machine (SVM) is utilized to classify feature sets extracted with 1D-CNN. The obtained results show that the features extracted in this method have excellent quality for fault classification without any additional feature selection.

Keywords: Gearbox, 1D-CNN, SVM, Fault Detection

1. INTRODUÇÃO

A exigência de alto rendimento em processos industriais torna necessária a adoção de ferramentas de detecção e diagnóstico de falhas rápidas, precisas e eficazes, evitando danos em máquinas ou equipamentos, o que, por consequência, pode prejudicar ou comprometer um processo (Park *et al.*, 2020).

Nesse contexto, o monitoramento de vibrações de máquinas e equipamentos industriais é prática comum de manutenção preditiva, sendo uma ferramenta importante para detecção e predição de falhas. Aquisição de dados, extração de parâmetros e classificação são fases gerais de uma gama de sistemas de monitoramento especializados, como mostrado em Fig. 1. Um dos procedimentos mais importantes no diagnóstico de falhas é a extração de atributos feita por métodos de processamento de sinais ou métodos estatísticos. Existem várias técnicas especializadas em extração de parâmetros. Entre os métodos de processamento de sinais, costuma-se dividi-los em técnicas no domínio do tempo (Samanta, 2004; Coraça and Dias Jr., 2018), domínio da frequência (Minamihara *et al.*, 1990) e tempo-frequência (Wang *et al.*, 2017). Recentemente as análises tempo-frequência têm atraído mais atenção por serem mais efetivas na identificação de variações das frequências ao longo do tempo, de modo que as intensidades de componentes não estacionárias podem ser estudadas.

Redes neurais artificiais profundas (DNN, do inglês *Deep Neural Network*) têm sido aplicadas na detecção e diagnóstico automatizado de máquinas e equipamentos tratando-os como problemas de classificação com base em padrões aprendidos. Entretanto, as redes neurais apresentam limitações quanto à capacidade de generalização, dando espaço a modelos que podem se ajustar melhor aos dados de treinamento. Máquinas de Vetores de Suporte (SVM, do inglês *Support Vector Machines*), com base na Teoria do Aprendizado Estatístico proposta por Vapnik (1995), estão sendo aplicadas nas áreas de aprendizado de máquina, visão computacional e reconhecimento de padrões devido à alta precisão e, principalmente, devido à boa capacidade de generalização. A principal diferença entre DNNs e SVMs está nos princípios de minimização de risco utilizados na fase de aprendizagem (MELLO and Ponti, 2018). No caso de SVMs, a minimização de risco estrutural condiz com a minimização do limite superior no risco esperado, enquanto as DNNs ajustam-se por minimização do risco empírico, minimizando o erro nos dados de treinamento. A diferença de abordagem leva a uma melhor

capacidade de generalização das SVMs em comparação às DNNs, possibilitando a utilização das SVMs em aplicações de monitoramento de máquinas com alta confiabilidade.

Redes Neurais Convolucionais (CNN, do inglês *Convolutional Neural Network*) fazem parte do acervo de técnicas de inteligência artificial com inspirações biológicas (LeCun *et al.*, 1989; Hubel and Wiesel, 1959). De acordo com os sinais de entrada, a rede pode aprender filtros de forma orientada aos dados e extrair atributos que descrevem as entradas. Durante a fase de treinamento supervisionado, os filtros convolucionais da CNN (*kernels*) são otimizados via algoritmos de retropropagação do erro (BP do inglês *Back-Propagation*) e, portanto, características altamente discriminativas podem ser aprendidas a partir dos dados de entrada. Além disso, as últimas camadas do classificador CNN são camadas totalmente conectadas convencionais. Consequentemente, com a CNN treinada tem-se a capacidade de extração de atributos e classificação de forma direta.

Neste estudo, um método híbrido composto por acoplamento de camadas convolucionais extratoras de atributos e máquina de vetor de suporte é utilizado na predição de danos via análise de vibrações de uma caixa de engrenagem com diferentes condições do engrenamento.

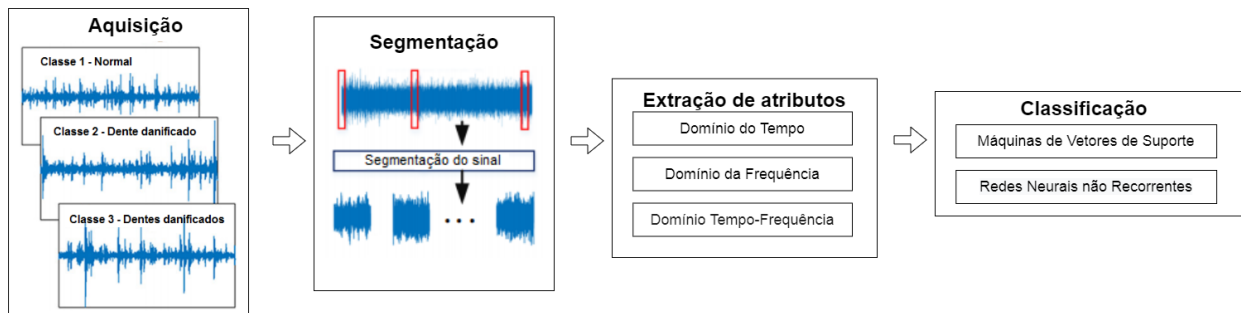


Figura 1: Fluxograma de processamento de sinais para detecção de falhas.

2. REDE NEURAL CONVOLUCIONAL

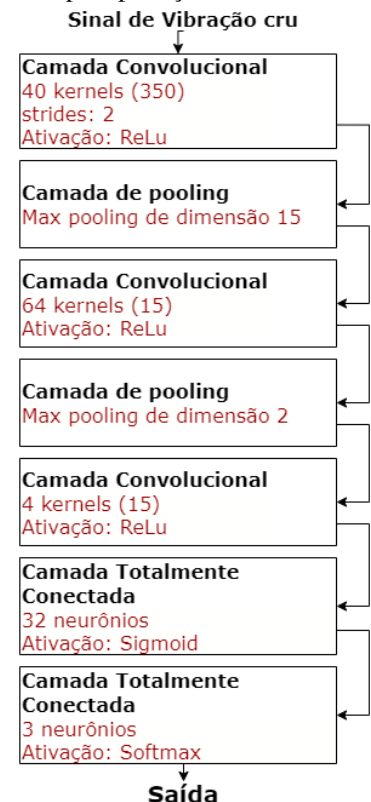
As Redes Neurais Convolucionais fornecem uma ferramenta poderosa para reconhecimento de padrões e classificação de sinais sem a necessidade de extração complicada de atributos. A arquitetura padrão da CNN geralmente consiste em três camadas importantes: convolucionais, *pooling* e totalmente conectadas. As camadas convolucionais extraem os atributos necessários para a classificação do sinal, enquanto as camadas de *pooling* visam reduzir a dimensão das entradas por meio de um operador de agrupamento (média dos dados para *average pooling* e máximo dos dados para *max pooling*) e, finalmente, as camadas totalmente conectadas são responsáveis pela classificação dos atributos extraídos. Normalmente, utiliza-se combinações de camadas convolucionais com camadas de *pooling* e, ao final da rede, camadas totalmente conectadas.

A arquitetura compacta das CNNs (em comparação às arquiteturas DNNs com múltiplas camadas totalmente conectadas), que realiza apenas convoluções, torna a ferramenta prática e econômica para monitoramento e detecção de falhas em tempo real. Este estudo propõe um modelo de CNN projetado especificamente para predição de falhas baseando-se no sinal de vibração de uma caixa de transmissão. O modelo combina camadas de *max pooling* e *strides* para compressão dos dados e extração de parâmetros do sinal. A arquitetura utilizada é representada na Fig. 2. O modelo consiste em três camadas convolutivas com *kernels* de dimensões 350, 15 e 15, respectivamente. Apenas foi inserido *stride* na primeira camada convolutiva. A primeira e a segunda camadas convolutivas são seguidas de camadas de *max pooling*, sendo a redução de um fator 8 entre a primeira e a segunda camadas convolucionais e um fator de 2 entre a segunda e a terceira camadas convolucionais.

Seja $\{s_i, y_i\} (i = 1, 2, \dots, 90)$ o conjunto de dados de treinos, com $s_i \in \mathbb{R}^{1250}$ o sinal segmentado de entrada e $y_i \in \{1, 2, 3\}$ a classe do segmento.

O objetivo da fase de treinamento de uma CNN é ajustar os pesos e os *bias* dos filtros de forma a minimizar o erro de classificação na camada de saída. A saída da m -ésima camada convolutiva pode ser representada por:

Figura 2: Arquitetura da CNN utilizada para predição de falhas.



$$\mathbf{S}_i^{(m)} = f(\mathbf{W}_C^{(m)} * \mathbf{S}_i^{(m-1)} + \mathbf{b}_C^{(m)}) . \quad (1)$$

O símbolo $*$ refere-se ao operador de convolução, $\mathbf{S}_i^{(m)}$ é a saída da m -ésima camada convolutiva para a amostra \mathbf{s}_i , os parâmetros $\mathbf{W}_C^{(m)}$ e $\mathbf{b}_C^{(m)}$ são os pesos do *kernel* e o *bias* da m -ésima camada convolutiva (o subscrito C é referente à camada convolucional), e $f(\cdot)$ é uma função de ativação não linear.

Dentre as funções de ativação, a função ReLU (do inglês, *Rectified linear units*) tem ganhado espaço em aplicações de CNNs, uma vez que demonstrou fornecer poderosa habilidade de modelar a não linearidade dos problemas (Bengio *et al.*, 1994). A função remove valores negativos, os definindo a zero.

$$f(z) = \max(0, z) . \quad (2)$$

A função ReLU evita saturações do gradiente do erro durante a fase de aprendizagem, além de exigir menor poder computacional quando comparado a outras funções de ativação não lineares. Dessa forma, a função é utilizada em todas as camadas convolucionais no presente trabalho.

As camadas totalmente conectadas, utilizadas nas camadas mais próximas a saída da rede neural, são responsáveis pela classificação dos atributos coletados. A saída da c -ésima camada totalmente conectada pode ser representada por:

$$\hat{\mathbf{y}}_i^{(c)} = f(\mathbf{z}_i^{(c)}) = f(\mathbf{W}_F^{(c)} \cdot \hat{\mathbf{y}}_i^{(c-1)} + \mathbf{b}_F^{(c)}) , \quad (3)$$

sendo que $\hat{\mathbf{y}}_i^{(c)}$ é a saída da c -ésima camada totalmente conectada para a amostra \mathbf{x}_i , os parâmetros $\mathbf{W}_F^{(c)}$ e $\mathbf{b}_F^{(c)}$ são os pesos sinápticos e o *bias* da c -ésima camada (o subscrito F é referente à camada totalmente conectada), e $f(\cdot)$ é uma função de ativação não linear. Foram utilizadas as funções de ativação sigmoide e *softmax*. A camada totalmente conectada de saída tem como função de ativação a função *softmax*, de forma que as saídas podem ser compreendidas como as probabilidades de que o sinal segmentado de entrada faça parte de cada uma das classes:

$$\begin{bmatrix} P(y_i = 1 | \hat{\mathbf{y}}_i^{(c-1)}; \mathbf{W}_F^{(c)}, \mathbf{b}_F^{(c)}) \\ P(y_i = 2 | \hat{\mathbf{y}}_i^{(c-1)}; \mathbf{W}_F^{(c)}, \mathbf{b}_F^{(c)}) \\ P(y_i = 3 | \hat{\mathbf{y}}_i^{(c-1)}; \mathbf{W}_F^{(c)}, \mathbf{b}_F^{(c)}) \end{bmatrix} = \frac{1}{\sum_{k=1}^3 \exp(z_{i,k})} \begin{bmatrix} \exp(z_{i,1}) \\ \exp(z_{i,2}) \\ \exp(z_{i,3}) \end{bmatrix} , \quad (4)$$

sendo $z_{i,k}$ é o k -ésimo elemento do vetor \mathbf{z}_i .

Utilizar como função custo a função de entropia cruzada em vez da soma dos quadrados dos erros leva a um treinamento mais rápido, bem como a uma generalização aprimorada (Bishop, 2006). Dessa forma, a entropia cruzada de multiclases é utilizada como função custo para obtenção do conjunto de parâmetros do modelo. O modelo é treinado pela minimização da função custo. Conforme o seguinte problema de otimização:

$$\min_{\Theta} L(\Theta) = -\frac{1}{90} \sum_{i=1}^N \sum_{j=1}^3 y_{i,j} \log[g_j(\Theta, \mathbf{s}_i)] , \quad (5)$$

em que $\Theta = \{\mathbf{W}_C^{(1)}, \mathbf{W}_C^{(2)}, \dots, \mathbf{b}_C^{(1)}, \mathbf{b}_C^{(2)}, \dots, \mathbf{W}_F^{(1)}, \mathbf{W}_F^{(2)}, \dots, \mathbf{b}_F^{(1)}, \mathbf{b}_F^{(2)}\}$ são os parâmetros da CNN a serem ajustados, sendo $y_{i,j} \in \{0, 1\}$ elemento da matriz *one hot* \mathbf{Y} , que apresenta a classe de cada segmento \mathbf{s}_i de forma binária, e $g_j(\Theta, \mathbf{s}_i)$ corresponde à j -ésima saída da função *softmax* para o sinal de entrada \mathbf{s}_i .

Os pesos Θ são ajustados utilizando o algoritmo de otimização baseado em gradiente descendente Adam, com hiperparâmetros escolhidos conforme o recomendado em (Kingma and Ba, 2014). O algoritmo Adam, amplamente utilizado no campo de aprendizado de redes neurais profundas, requer o cálculo do vetor gradiente ∇L , que pode ser obtido com a retropropagação do erro (BP). Mais detalhes do algoritmo BP podem ser encontrados no trabalho de Abdeljaber *et al.* (2017).

3. MÁQUINA DE VETORES DE SUPORTE

Fundamentada pela Teoria do Aprendizado Estatístico (Vapnik, 1995), as técnicas de aprendizagem de Máquinas de Vetores de Suporte (SVM) buscam a minimização do Risco Estrutural (SRM, do inglês *Structural Risk Minimization*) ao levar em conta o desempenho do classificador no conjunto de treinamento e a complexidade do classificador.

As Máquinas de Vetores de Suporte com diferentes funções *kernel* podem transformar um problema não-linearmente separável em um problema linearmente separável ao projetar os dados num espaço de dimensão maior e então encontrar o hiperplano separador ideal (conforme ilustra a Fig. 3). Este método foi inicialmente proposto para resolver problemas binários (de duas classes). Mais tarde, algumas estratégias foram sugeridas para estender esta técnica para problemas de classificação de multiclases (Bottou *et al.*, 1994; Knerr *et al.*, 1990).

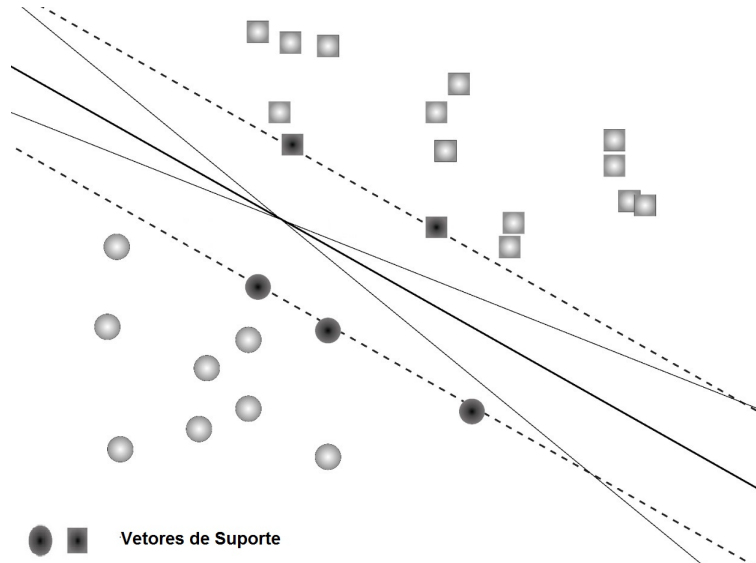


Figura 3: Hiperplano de separação ótimo (linha grossa), com os seus hiperplanos de suporte (tracejados). Outros hiperplanos separadores possíveis em finas linhas contínuas.

3.1 HIPERPLANO ÓTIMO PARA PROBLEMAS DE CLASSIFICAÇÃO BINÁRIO

Considere um determinado conjunto D de dados observados com m exemplos para um problema de classificação binária,

$$D = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)\} \in X \times Y, \quad (6)$$

sendo $\mathbf{x} \in X = \mathbb{R}^n$ o espaço das entradas e $y \in Y = \{-1, +1\}$ uma classe ou rótulo.

A SVM busca encontrar o hiperplano de margem máxima onde os dados de treinamento \mathbf{x}_i , mapeados para um espaço de dimensão maior pela função $\phi(\cdot)$, são então linearmente separados de acordo com a solução do seguinte problema de otimização primal:

$$\begin{aligned} &\text{Minimizar } \Gamma(\mathbf{w}, b, \xi) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^m \xi_i \\ &\text{Sujeito a } \begin{cases} \forall_{i=1}^m : y_i(\langle \phi(\mathbf{x}_i), \mathbf{w} \rangle + b) \geq 1 - \xi_i \\ \forall_{i=1}^m : \xi_i \geq 0 \end{cases} \end{aligned} \quad (7)$$

em que $\mathbf{w} \in \mathbb{R}^n$ é o vetor normal ao hiperplano classificador, b é um escalar, ξ é a folga das variáveis, C é o parâmetro de penalidade que controla o compromisso entre a maximização da margem e a minimização dos erros de classificação. A Fig. 4 apresenta os hiperplanos ótimos de um mesmo problema de classificação para diferentes parâmetros C .

Para encontrar a solução do problema de otimização primal dado pelas Eqs. (7), geralmente se recorre ao seu respectivo problema de otimização dual obtido com o método dos multiplicadores de Lagrange:

$$\begin{aligned} &\text{Maximizar } W(\alpha) = -\frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) + \sum_{i=1}^m \alpha_i \\ &\text{Sujeito a } \begin{cases} \forall_{i=1}^m : 0 \leq \alpha_i \leq C \\ \sum_{i=1}^m \alpha_i y_i = 0 \end{cases} \end{aligned} \quad (8)$$

sendo $\alpha = (\alpha_1, \dots, \alpha_m)$ a variável dual ou o vetor dos multiplicadores de Lagrange. A função *kernel* $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$ calcula o produto interno dos dados de entrada no espaço de dimensão maior sem a necessidade do mapeamento prévio $\phi(\cdot)$. Esse método é conhecido como o truque do *kernel* (*kernel trick*). A função de decisão do classificador é dada por:

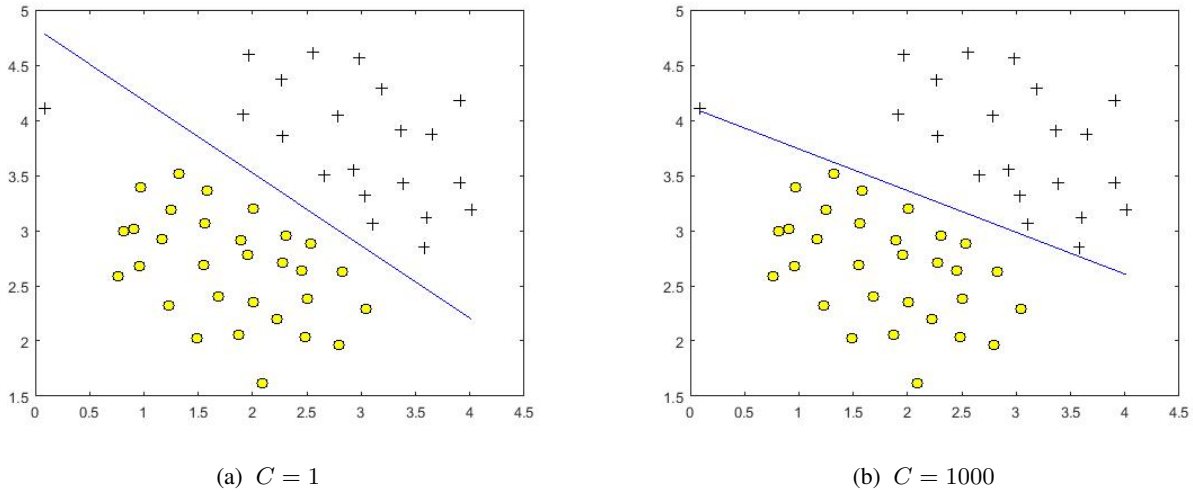


Figura 4: Hiperplanos de separação ótimos para diferentes parâmetros de penalidade C .

$$g(\mathbf{x}) = \text{sign} \left[\sum \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b \right], \quad (9)$$

onde os pontos \mathbf{x}_i são vetores de suporte.

Utilizou-se e comparou-se os resultados de SVMs com duas funções kernels:

Kernel linear: $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$,

Kernel de base radial: $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$.

(10)

3.2 SVM PARA MULTICLASSES

Dois abordagens são utilizadas para resolução do problema de classificação de múltiplas classes com SVM. Uma abordagem baseia-se na construção e combinação de várias SVMs de duas classes, três estratégias típicas são aplicadas: "Um-Contra-Todos", "Um-Contra-Um" e "DAGSVM" (do inglês *Directed Acyclic Graph Support Vector Machines*) que quer dizer SVM com Grafo Direcionado Acíclico. A outra abordagem considera todos os dados multiclases em um único problema de otimização. Hsu and Lin (2002) compararam essas duas abordagens em conjuntos de dados diferentes e concluíram que a abordagem com combinações de SVMs, especificamente, Um-Contra-Um e DAGSVM são os métodos mais adequados.

O presente trabalho utilizou-se da biblioteca scikit-learn (Pedregosa *et al.*, 2011) para desenvolvimento das SVMs. A scikit-learn é uma biblioteca de código aberto para programação em Python eficiente para problemas de classificação e regressão. A biblioteca possui a opção de utilização do método Um-Contra-Um para SVMs multiclases. Considerando um problema com k classes existentes, o método Um-Contra-Um necessita da construção de $k(k-1)/2$ classificadores, onde cada classificador usa os dados de treinamento de duas classes.

4. MODELO HÍBRIDO CNN-SVM

A arquitetura do modelo híbrido CNN-SVM foi projetada substituindo as camadas totalmente conectadas da CNN por uma SVM. Embora os parâmetros dos *kernels* utilizados para extração de atributos pelas camadas convolucionais sejam treinados em conjunto com as camadas totalmente conectadas, não há a necessidade de preservar a arquitetura padrão da CNN ao passo que uma SVM tende a ter maior capacidade de generalização. Os atributos extraídos pelas camadas convolucionais podem ser tratados como recursos para qualquer outro classificador.

A Fig. 5 apresenta a arquitetura do modelo híbrido adotado. A proposta do modelo híbrido se baseia no ideal de que o novo classificador compense os limites dos modelos CNN e SVM. Tem-se a expectativa de que o modelo híbrido CNN-SVM apresente desempenho superior com relação ao classificador CNN, ao incorporar os méritos de ambos os classificadores.

5. EXPERIMENTOS

O banco de dados utilizado no processo de treinamento e teste do modelo híbrido CNN-SVM é aberto à comunidade científica conforme mencionado em (Zamanian, 2010). Esse mesmo banco de dados também foi utilizado no desenvolvimento dos trabalhos (Zamanian and Ohadi, 2011, 2016). O sinal coletado está relacionado à vibração radial de uma caixa

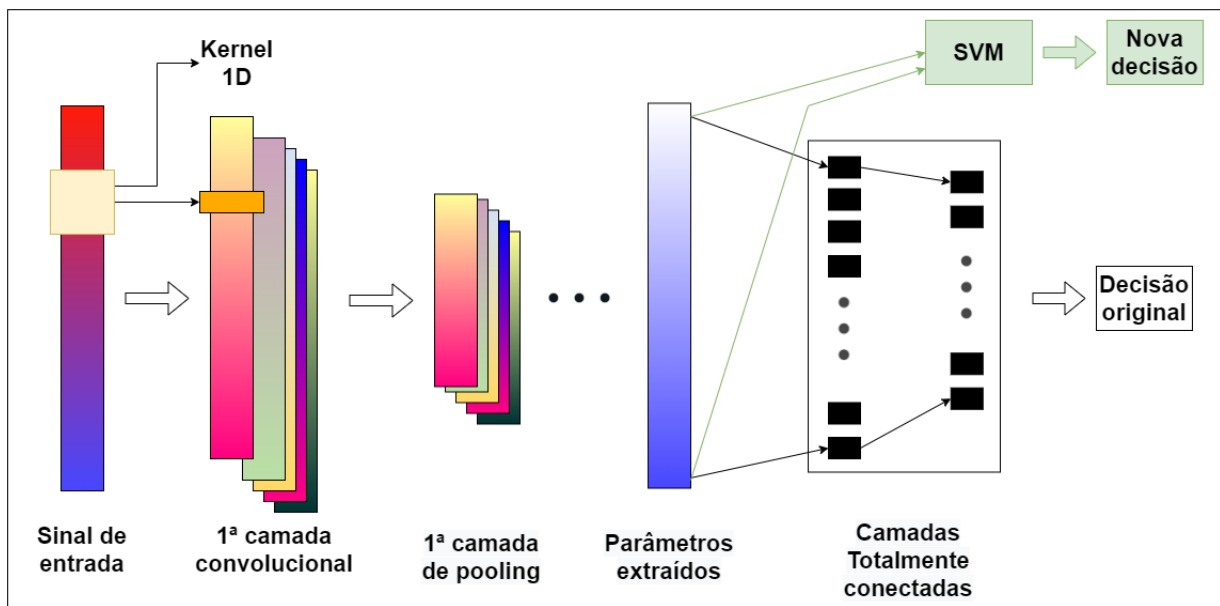
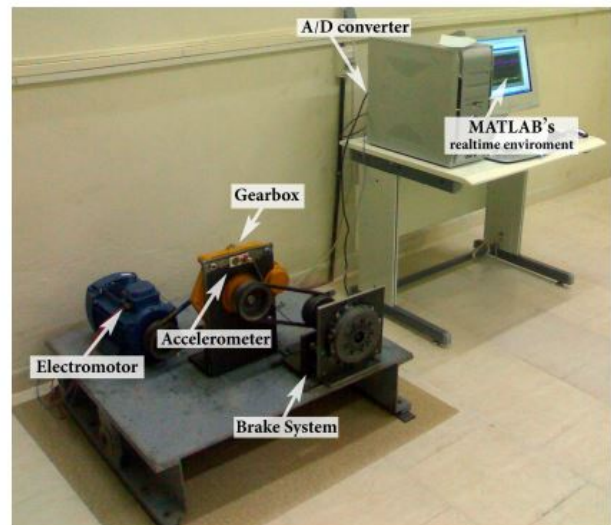


Figura 5: Estrutura do modelo de classificação híbrido CNN-SVM.

de engrenagem com três diferentes condições de engrenamento helicoidal (Fig. 6a): engrenagem saudável, engrenagem com três dentes desgastados e engrenagem com um dente lascado.



(a) Condições das engrenagens: (1) Engrenagem esquerda: Saudável (2) Engrenagem intermediária: Dentes desgastados (3) Engrenagem direita: Dente lascado [Figura extraída de (Zamanian and Ohadi, 2011)].



(b) Configuração de teste experimental [Figura extraída de Zamanian (2010)].

Figura 6: Condições de engrenamento (a) e setup experimental (b).

O engrenamento helicoidal com pinhão de 15 dentes e engrenagem de 110 dentes fornece relação de velocidades de 7,33. Em cada uma das três condições, o sinal foi coletado por 10 segundos com taxa de amostragem de 10 kHz. Mais detalhes sobre o procedimento de aquisição de dados e equipamentos utilizados podem ser encontrados em Zamanian (2010). Os testes foram realizados a uma rotação nominal do pinhão de 1440 rpm. O setup utilizado pode ser visualizado conforme a Fig. 6b.

6. RESULTADOS E DISCUSSÕES

Para avaliar a eficiência do modelo híbrido CNN-SVM no diagnóstico de falhas de aplicações práticas, utilizou-se os classificadores CNN e CNN-SVM com *kernel* linear e *kernel* de base radial (RBF, do inglês *Radial Basis Function*) aos dados de vibrações coletados. O classificador CNN com arquitetura representada pela Fig. 2 foi desenvolvido com a utilização da biblioteca de código aberto TensorFlow (Abadi *et al.*, 2015), especializada em métodos de aprendizado de máquinas para a linguagem de programação Python.

Para cada uma das três condições de engrenamento, os dados foram amostrados por 10 s. Conforme a segmentação realizada em (Zamanian and Ohadi, 2011), cada sinal coletado foi dividido em 80 segmentos com o comprimento de 1250 amostras. No total, 240 (80×3) sinais segmentados foram utilizados, sendo que, destes, 90 segmentos fizeram parte do conjunto de treino utilizados para ajuste das máquinas de aprendizado, e os 150 segmentos restantes foram utilizados para testar o desempenho dos modelos.

Os atributos extraídos pela CNN foram utilizados para treinamento de duas SVMs. Uma com *kernel* linear e parâmetro $C = 1$ e uma segunda com *kernel* RBF com $\gamma = 0,00044$ (parâmetros definidos de acordo com o padrão da biblioteca scikit-learn).

O classificador CNN com arquitetura representada pela Fig. 2 foi treinada com o algoritmo Adam em 35 épocas. A Fig. 7 ilustra os resultados do modelo CNN sobre o conjunto de teste.

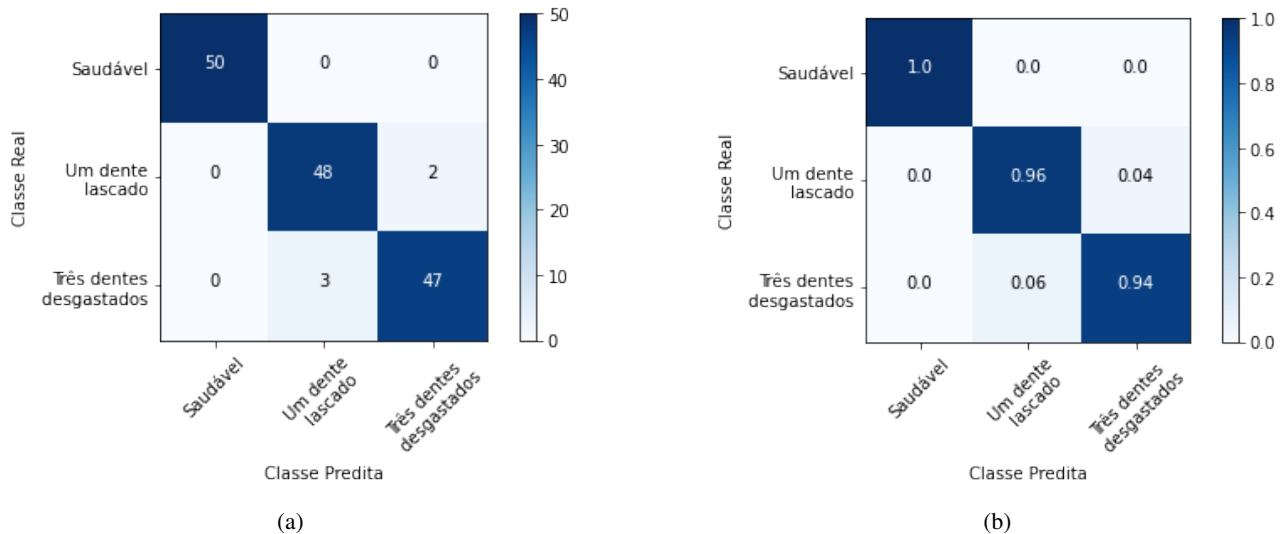


Figura 7: Matrizes de Confusão do modelo CNN aplicado aos dados de teste. Resultado real (a) e normalizado (b).

Os atributos extraídos do conjunto de treino foram utilizados na fase de aprendizado das SVMs. As Fig. 8 e 9 apresentam os resultados das SVMs sobre o conjunto de teste. Os melhores resultados dos modelos híbridos CNN-SVM quando comparado ao classificador CNN são condizentes com a expectativa de que o modelo híbrido possa incorporar os méritos dos classificadores CNN e SVM. Os agrupamentos observados na Fig. 10 corroboram a afirmação de que as camadas convolucionais são ótimas extratoras de atributos.

Os atributos extraídos pelas camadas convolucionais são pertencentes a um espaço de dimensão 28. Como forma de visualização dos dados no espaço de dimensão 28, é possível utilizar o algoritmo t-SNE (do inglês, *T-distributed Stochastic Neighbor Embedding*). O método busca apresentar a distribuição dos pontos de um espaço de dimensão maior num espaço 2D ou 1D preservando as relações de distanciamento entre os pontos (Maaten and Hinton, 2008). O resultado desse algoritmo tornam ainda maiores as distâncias euclidianas de pontos no espaço visualizado cuja distância no espaço de dimensão maior seja elevada. Dessa forma, este é um bom método de apresentação de dados. A Fig. 10 apresenta os dados de teste após extração de atributos pelas camadas convolucionais e utilização do algoritmo t-SNE.

7. CONCLUSÃO

Os atributos extraídos pelas camadas convolucionais mostraram-se suficientes para distinção dos dados por parte dos classificadores. Os modelos híbridos CNN-SVM apresentaram melhores resultados quando comparado ao classificador CNN, corroborando com a expectativa de que o modelo híbrido incorpore qualidades dos modelos CNN e SVM.

8. AGRADECIMENTOS

Agradecimento especial a Natan Goes da empresa Spotlight pelas contribuições com ilustrações.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES)

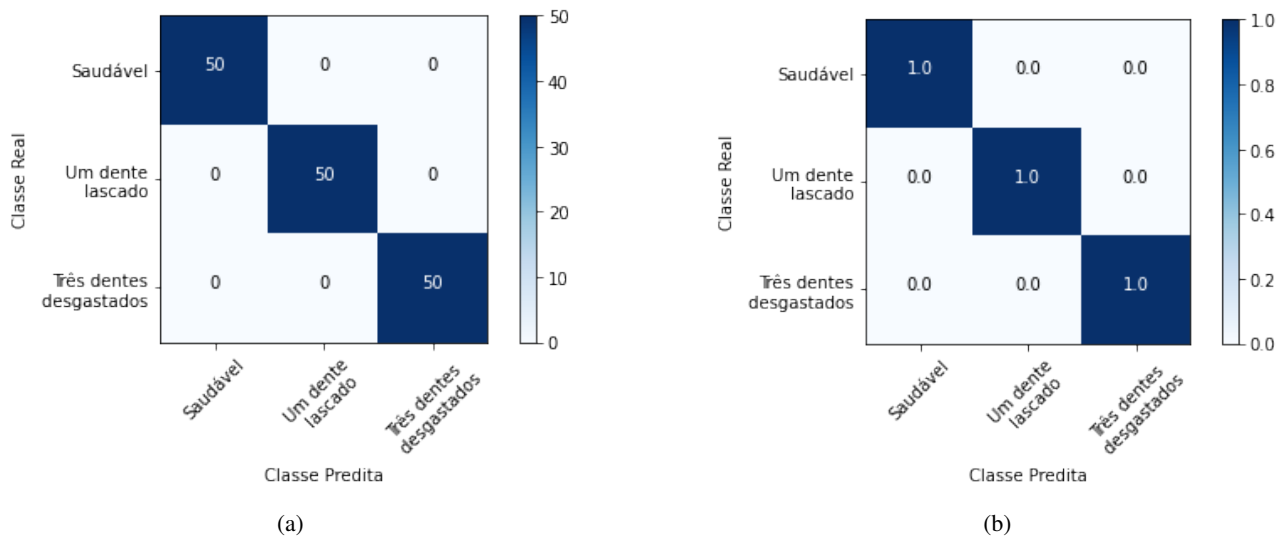


Figura 8: Matrizes de Confusão do modelo CNN-SVM com *kernel* linear aplicado aos dados de teste. Resultado real (a) e normalizado (b).

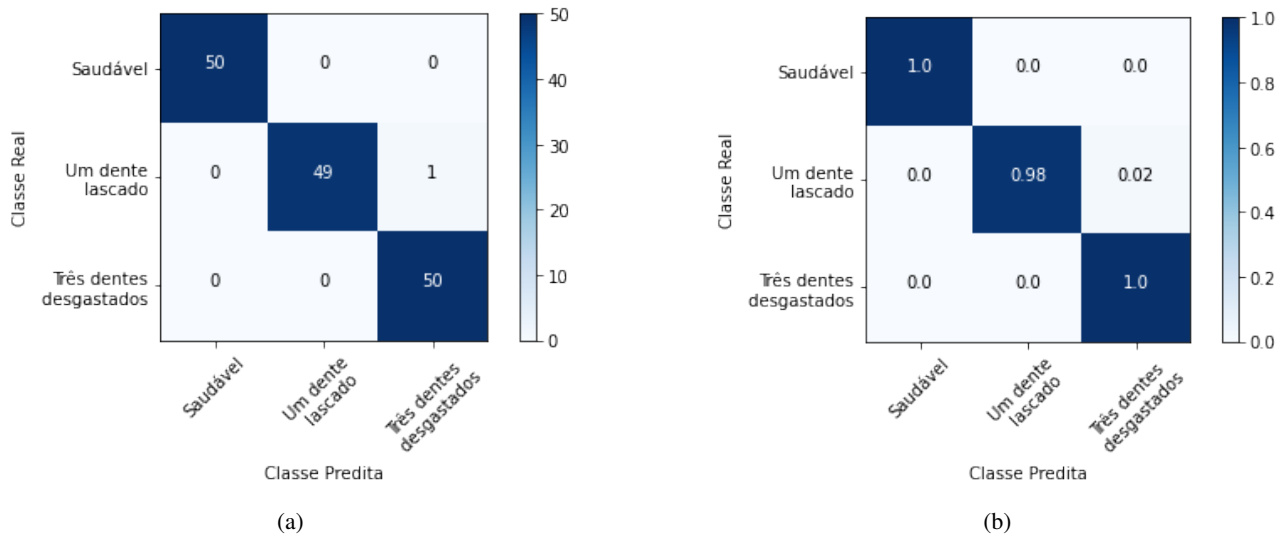


Figura 9: Matrizes de Confusão do modelo CNN-SVM com *kernel* RBF aplicado aos dados de teste. Resultado real (a) e normalizado (b).



Figura 10: Representação dos atributos dos dados de teste no \mathbb{R}^2 pelo algoritmo t-SNE.

9. REFERÊNCIAS

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y. and Zheng, X., 2015. "TensorFlow: Large-scale machine learning on heterogeneous systems". URL <https://www.tensorflow.org/>. Software available from tensorflow.org.
- Abdeljaber, O., Avci, O., Kiranyaz, S., Gabbouj, M. and Inman, D.J., 2017. "Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks". *Journal of Sound and Vibration*, Vol. 388, pp. 154–170.
- Bengio, Y., Simard, P. and Frasconi, P., 1994. "Learning long-term dependencies with gradient descent is difficult". *IEEE Transactions on Neural Networks*, Vol. 5, No. 2, pp. 157–166. doi:10.1109/72.279181.
- Bishop, C.M., 2006. *Pattern recognition and machine learning*. springer.
- Bottou, L., Cortes, C., Denker, J.S., Drucker, H., Guyon, I., Jackel, L.D., LeCun, Y., Muller, U.A., Sackinger, E., Simard, P. et al., 1994. "Comparison of classifier methods: a case study in handwritten digit recognition". In *Proceedings of the 12th IAPR International Conference on Pattern Recognition, Vol. 3-Conference C: Signal Processing (Cat. No. 94CH3440-5)*. IEEE, Vol. 2, pp. 77–82.
- Coraça, E.M. and Dias Jr, M.D., 2018. "Model-based identification of rotor-bearing system parameters employing adaptive filtering". In *International Conference on Rotor Dynamics*. Springer, pp. 236–249.
- Hsu, C.W. and Lin, C.J., 2002. "A comparison of methods for multiclass support vector machines". *IEEE transactions on Neural Networks*, Vol. 13, No. 2, pp. 415–425.
- Hubel, D.H. and Wiesel, T.N., 1959. "Receptive fields of single neurones in the cat's striate cortex". *The Journal of physiology*, Vol. 148, No. 3, p. 574.
- Kingma, D.P. and Ba, J., 2014. "Adam: A method for stochastic optimization". *arXiv preprint arXiv:1412.6980*.
- Knerr, S., Personnaz, L. and Dreyfus, G., 1990. "Single-layer learning revisited: a stepwise procedure for building and training a neural network". In *Neurocomputing*, Springer, pp. 41–50.
- LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W. and Jackel, L.D., 1989. "Backpropagation applied to handwritten zip code recognition". *Neural computation*, Vol. 1, No. 4, pp. 541–551.
- Maaten, L.v.d. and Hinton, G., 2008. "Visualizing data using t-sne". *Journal of machine learning research*, Vol. 9, No. Nov, pp. 2579–2605.
- MELLO, R.F. and Ponti, M.A., 2018. *Machine Learning: A Practical Approach on the Statistical Learning Theory*. Springer.
- Minamihara, H., Nishimura, M., Takakuwa, Y. and Ohta, M., 1990. "A method of detection of the correlation function and frequency power spectrum for random noise or vibration with amplitude limitation". *Journal of sound and vibration*, Vol. 141, No. 3, pp. 425–434.
- Park, Y.J., Fan, S.K.S. and Hsu, C.Y., 2020. "A review on fault detection and process diagnostics in industrial processes". *Processes*, Vol. 8, No. 9, p. 1123.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E., 2011. "Scikit-learn: Machine learning in Python". *Journal of Machine Learning Research*, Vol. 12, pp. 2825–2830.
- Samanta, B., 2004. "Gear fault detection using artificial neural networks and support vector machines with genetic algorithms". *Mechanical systems and signal processing*, Vol. 18, No. 3, pp. 625–644.
- Vapnik, V.N., 1995. *The Nature of Statistical Learning Theory*. Springer-Verlag, Berlin, Heidelberg. ISBN 0387945598.
- Wang, P., Yan, R., Gao, R.X. et al., 2017. "Virtualization and deep recognition for system fault classification". *Journal of Manufacturing Systems*, Vol. 44, pp. 310–316.
- Zamanian, A.H., 2010. "Experimental dataset for gear fault diagnosis".
- Zamanian, A.H. and Ohadi, A., 2011. "Gear fault diagnosis based on gaussian correlation of vibrations signals and wavelet coefficients". *Applied Soft Computing*, Vol. 11, No. 8, pp. 4807–4819.
- Zamanian, A.H. and Ohadi, A., 2016. "Gearbox fault detection through pso exact wavelet analysis and svm classifier". *arXiv preprint arXiv:1605.04874*.

10. RESPONSABILIDADE PELAS INFORMAÇÕES

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