

Detecção de Efeito Corona em Isoladores Poliméricos de Alta Tensão via Redes Neurais Artificiais

Danilo Avilar Silva*, Ajalmar R. Rocha Neto[†], Elias T. da Silva Júnior[‡],
Francisco J. A. de Aquino[§], André Luis C. de Araújo[¶]
Instituto Federal do Ceará (IFCE)

André D. Klein^{||}, Fredson M. da Silva^{**}
ELETROBRAS Distribuição Acre

Email: danilo.avilar@ifce.edu.br*, ajalmar@ifce.edu.br[†], elias@ifce.edu.br[‡], fcoalves_aq@ifce.edu.br[§],
andreluiz.ifce@gmail.com[¶], andre.klein@eletrobrasacre.com^{||}, fredson.silva@eletrobrasacre.com^{**}

Resumo—A ocorrência de efeito corona é um importante indicativo de que um isolador apresenta ou pode vir a apresentar defeitos. Assim, a detecção de efeito corona pode auxiliar no diagnóstico de problemas em isoladores, situação que pode ocasionar sérios danos ao sistema, tal como a interrupção repentina do serviço. Redes neurais artificiais (RNA) são técnicas de aprendizado de máquina que permitem classificar padrões em categorias. Uma outra técnica interessante que pode ser aplicada na resolução deste tipo de problema são as *support vector machines*. Neste trabalho é avaliada a capacidade de aplicação de redes neurais para a detecção de efeito corona em isoladores poliméricos de alta tensão a partir de informações de ultrassom. Com base nos resultados obtidos, pode-se inferir que as redes neurais apresentaram alta capacidade de generalização, mostrando-se uma técnica bastante promissora neste tipo de problema.

Index Terms—redes neurais artificiais, perceptron multicaudadas, *support vector machines*, efeito corona, monitoramento, diagnóstico.

I. INTRODUÇÃO

Falhas em isoladores elétricos de média e alta tensão ocasionadas por problemas como poluição, fissuras, falhas estruturais, entre outros, são caracterizadas pela emissão de efeitos elétricos e sônicos que podem ser observados respectivamente por circuitos de captação de radiação eletromagnética e circuitos de detecção de ruídos ultrassônicos [1].

Os fenômenos elétricos e sônicos ocorrem de forma simultânea, de tal maneira que os seguintes fenômenos podem ocorrer [1], a saber:

- Arco elétrico, esse fenômeno ocorre quando existe um fluxo de corrente elétrica através de espaçamentos em ar, como exemplo pode-se citar uma descarga atmosférica;
- Efeito corona, esse fenômeno ocorre quando o gradiente de tensão na superfície de um condutor elétrico, como uma linha aérea de distribuição de alta tensão, excede o valor limite (denominado gradiente crítico); e
- Trilhamento elétrico (*tracking*), esse fenômeno ocorre quando existem descargas incompletas, de caráter repeti-

tivo, sobre superfícies isolantes. Em materiais orgânicos, normalmente esse fenômeno resulta em caminhos condutores devido à degradação associada com descargas contínuas e erosão.

Estes fenômenos podem ser influenciados pela presença de umidade, chuva, salinidade do isolador, estado de poluição, variação da temperatura ou pressão atmosférica, dentre outros. Todos esses efeitos podem ser simulados em laboratório e medidos simultaneamente por equipamentos de captação de rádio frequência, medidores de ultrassom, termovisores e isolômetros, utilizados na inspeção instrumental das redes de distribuição. O grande problema verificado, é que parte destes equipamentos foi desenvolvido para outras finalidades, principalmente para a aplicação em indústrias. A experiência tem mostrado que diferentes observadores produzem diagnósticos totalmente diferentes, elevando o erro no procedimento e tornando os resultados não confiáveis e, assim, levando à substituição de isoladores bons, bem como à permanência de dispositivos ruins na rede. Além disso, há uma antecipação de investimentos, gastos com mão de obra e em muitos casos a não solução dos problemas [2].

Novas técnicas têm sido propostas com o intuito de auxiliar no procedimento de detecção desses fenômenos [3], [4]. Uma técnica bastante interessante que também vem sendo aplicada a esse tipo de problema são as redes neurais artificiais (RNA) [5]. Em Bezerra *et. al* [6], uma rede neural artificial foi utilizada para realizar o diagnóstico de defeitos (ou não) em isoladores de subestações e de linhas de transmissão/distribuição. Porém, os resultados apresentados neste trabalho não foram conclusivos, principalmente no que se refere à capacidade de generalização do processo de tomada de decisão. Em outro trabalho relacionado, Capelini [7] utiliza redes neurais com o intuito de reconhecer padrões de formas de onda de descargas parciais, permitindo assim sua associação com o ponto de ocorrência do efeito corona.

Sendo assim, em decorrência do exposto e da já conhecida

aplicabilidade de RNA em tarefas de classificação de padrões nos mais diversos tipos de problemas do mundo real, neste trabalho tem-se por objetivo detectar defeitos em isoladores com base nos ruídos emitidos pelos mesmos em operação. A importância desse trabalho é de fácil verificação, pois uma ferramenta de diagnóstico como esta pode reduzir a ocorrência de interrupção no fornecimento de energia elétrica. Mais especificamente, neste trabalho as RNA são utilizadas para detectar a presença ou ausência de efeito corona a partir de ruídos ultrassônicos provenientes de isoladores de alta tensão.

O restante deste artigo está organizado como descrito a seguir. Na Seção II são apresentadas as arquitetura avaliadas da detecção de efeito corona, bem como uma breve introdução as RNA. Já na Seção III são apresentados os resultados obtidos. Por fim, na Seção IV, são apresentadas as conclusões alcançadas a partir dos experimentos.

II. REDES NEURAIS ARTIFICIAIS

Redes Neurais Artificiais são sistemas paralelos distribuídos (normalmente não-lineares) compostos por unidades de processamento que têm como base o neurônio biológico [8]. Estas unidades de processamento são denominadas neurônios artificiais e são compostas por terminais de entradas (dendritos), que recebem os valores de entrada do padrão a ser analisado. Para representar as intensidades das conexões entre os neurônios (ligações sinápticas), cada entrada tem um vetor peso associado responsável pelo grau de acoplamento sináptico entre um neurônio que envia o sinal e outro que recebe. A sinapse é dita excitatória se o vetor peso tem valor positivo e inibitória se o vetor peso possuir valor negativo. Em termos de modelagem, existe uma entrada adicional denominada de viés (*bias*) que visa modelar o limiar de disparo do neurônio.

Segundo o modelo matemático de McCulloch e Pitts [9], a saída de um neurônio é obtido pela soma ponderada dos valores de entrada pelos seus respectivos pesos, tal como:

$$v = w_1x_1 + w_2x_2 + \dots + w_mx_m - \theta \quad (1)$$

$$= \sum_{i=1}^p w_i x_i - \theta, \quad (2)$$

em que x_i são os sinais de entrada, w_i os pesos sinápticos, p o número de entradas, θ o bias e v a saída da combinação linear.

A saída do combinador linear é apresentada a uma função de ativação $\phi(\cdot)$ que limita o valor de saída em um intervalo desejado. Estas funções podem apresentar características lineares e não lineares. Entretanto, as funções de ativação não-lineares apresentam uma melhor aproximação ao funcionamento do neurônio biológico, sendo a função sigmoide uma das mais utilizadas [10].

A capacidade de aprendizado das RNA ocorre por meio de um processo iterativo denominado de algoritmo de aprendizagem (treinamento), que consiste em constantes ajustes aplicados aos vetores de pesos até que um critério de desem-

penho preestabelecido seja alcançado [8]. Uma regra típica de aprendizado é apresentada a seguir.

$$w_i(t+1) = w_i(t) + \eta e(t)x_i(t) \quad (3)$$

em que $\eta \in [0,1]$ é a taxa de aprendizagem, $w_i(t+1)$ e $w_i(t)$ são os valores da i -ésima componente do vetor de peso nas iterações (tempos) t (atual) e $t+1$ (novo), respectivamente; bem como $x_i(t)$ é o valor da i -ésima componente do vetor de entrada. Além disto, $e(t)$ é o erro obtido na i -ésima iteração, tal que $e(t) = d(t) - v(t)$, em que $d(t)$ é a saída desejada e $v(t)$ é a saída do neurônio. Vale ressaltar que a combinação do modelo de McCulloch e Pitts com a regra de aprendizagem descrita acima foi apresentada no trabalho sobre o Perceptron Simples [11].

Como dito, o algoritmo de aprendizagem depende do modelo de rede empregado e deve estabelecer um equilíbrio entre a capacidade de generalização e a complexidade do modelo. Um mau dimensionamento pode resultar em um sobreajuste da rede aos dados utilizados para treinamento (*overfitting*) ou ainda, fazer com que a rede não seja treinada suficientemente para produzir resultados (*underfitting*).

A seguir são apresentadas as redes de interesse para esse trabalho, a saber: Perceptron Multicamadas (*Multilayer Perceptron*, MLP) e as máquinas de vetores-suporte (*Support Vector Machines*, SVM).

A. Rede Perceptron Multicamadas

Uma rede neural perceptron com várias camadas (denominada Rede Perceptron Multicamadas) é constituída de uma camada de entrada que recebe os vetores de atributos, uma ou mais camadas intermediárias, compostas por neurônios somadores com função de ativação não-linear e uma camada de saída, também composta por neurônios somadores (que podem ser lineares ou não) [8].

Nas definições e cálculos mostrados a seguir, considera-se uma arquitetura de rede neural MLP com apenas uma camada escondida de neurônios, treinada com o algoritmo de retro propagação do erro (*error backpropagation algorithm*).

O vetor de pesos associado ao i -ésimo neurônio da camada escondida, também chamada de camada oculta ou camada intermediária, é descrito por

$$\mathbf{w}_i = \begin{pmatrix} w_{i0} \\ \vdots \\ w_{in} \end{pmatrix} = \begin{pmatrix} \theta_i \\ \vdots \\ w_{in} \end{pmatrix} \quad (4)$$

em que θ_i é o limiar associado ao i -ésimo neurônio.

De modo semelhante, o vetor de pesos associado ao k -ésimo neurônio da camada de saída é representado por

$$\mathbf{m}_k = \begin{pmatrix} m_{k0} \\ \vdots \\ m_{kn} \end{pmatrix} = \begin{pmatrix} \theta_k \\ \vdots \\ m_{kn} \end{pmatrix} \quad (5)$$

em que θ_k é o limiar associado ao k -ésimo neurônio de saída.

O treinamento da rede MLP se dá em dois sentidos: um direto e outro reverso. No sentido direto, o fluxo de sinais

(informação) se dá dos neurônios de entrada para os neurônios de saída, passando obviamente pelos neurônios da camada escondida.

Dessa forma, após a apresentação de um vetor de entrada $\mathbf{x}$, na iteração t , o primeiro passo a ser executado no treinamento é calcular as ativações dos neurônios da camada escondida.

$$u_i(t) = \sum_{j=0}^n w_{ij}x_j(t) = \mathbf{w}_i^T(t)\mathbf{x}(t) \quad (6)$$

$$v_i(t) = \varphi[u_i(t)] \quad (7)$$

Tal que a função de ativação φ , em geral, assume uma das seguintes formas:

$$\varphi(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \text{ (Tangente Hiperbólica)} \quad (8)$$

e

$$\varphi(x) = \frac{1}{1 + e^{-x}}, \text{ (Logística).} \quad (9)$$

O segundo passo consiste em repetir os computos apresentados nas equações 6 e 7 para cada um dos neurônios da camada de saída, i.e., calcular as ativações dos neurônios de saída, como segue:

$$u_k(t) = \sum_{j=0}^q m_{ki}v_j(t) = M_i^T(t)V(t) \quad (10)$$

$$y_i(t) = \varphi[u_k(t)] \quad (11)$$

No sentido reverso, o fluxo se dá dos neurônios de saída para os neurônios de entrada, passando pelos neurônios da camada escondida.

Assim, após os cálculos das ativações e saídas no sentido direto, o primeiro passo do sentido reverso consiste em calcular os gradientes locais $\delta_k(t)$ dos neurônios da camada de saída.

$$\delta_k(t) = e_k \varphi'[u_k(t)], k = 1, \dots, m \quad (12)$$

em que

$$e_k(t) = d_k(t) - y_k(t). \quad (13)$$

é o erro entre a saída desejada $d_k(t)$ e a saída obtida $y_k(t)$ no k -ésimo neurônio, respectivamente. Ressalta-se mais uma vez que a derivada $\varphi'[u_k(t)]$ assume diferentes expressões dependendo da escolha da função de ativação, como apresentado anteriormente, de tal maneira que uma escolha comum é a sigmoide logística ou tangente hiperbólica.

O segundo passo do sentido reverso consiste em calcular os gradientes locais

$$\delta_i(t) = \varphi'[u_i(t)] \sum_{k=1}^m m_{ki}(t)\delta_k(t), i = 1, \dots, q \quad (14)$$

dos neurônios da camada escondida.

O terceiro passo corresponde ao processo de atualização dos pesos da rede. Assim, para a camada escondida tem-se que a regra de atualização dos pesos, w_{ij} , é dada por

$$\begin{aligned} w_{ij}(t+1) &= w_{ij}(t) + \Delta w_{ij}(t) \\ &= w_{ij}(t) + \eta \delta_i(t) x_j(t) \end{aligned} \quad (15)$$

em que η é a taxa de aprendizagem.

Já para a camada de saída tem-se que a regra de atualização dos pesos m_{ki} é dada por

$$\begin{aligned} m_{ki}(t+1) &= m_{ki}(t) + \Delta m_{ki}(t) \\ &= m_{ki}(t) + \eta \delta_k(t) y_i(t) \end{aligned} \quad (16)$$

B. Support Vector Machines

Uma Máquina de Vetor-Suporte (*Support Vector Machines*, SVM) é uma generalização não linear do algoritmo proposto por Vapnik e Lerner [12], denominado *Generalized Portrait*. Posteriormente, foi proposta uma descrição do algoritmo para a construção de hiperplanos de separação com margem ótima [13]. Após décadas de pesquisas e com um forte embasamento na teoria de aprendizagem estatística [14] surgem as SVMs [15].

Conceitualmente, o funcionamento das SVMs se resume no mapeamento de vetores de entrada para um espaço de altíssima dimensão ($\mathbb{R}^n \rightarrow \mathbb{R}^{n_h}$). Neste espaço, as características para uma superfície de decisão linear são encontradas [16]. É considerado um método geral de aprendizagem supervisionado, aplicado tanto a problemas de classificação quanto de regressão. As SVMs baseiam-se no princípio de minimização do risco estrutural (*Structural Risk Minimization*, SRM) que busca a maximização da capacidade de generalização, o que resulta em modelos com desempenho superior ao das redes neurais convencionais, que empregam apenas a minimização do risco empírico (*Empirical Risk Minimization*, ERM) [17].

De um modo geral, o resultado do mapeamento realizado no processo de treinamento da SVM consiste em encontrar os padrões mais próximos ao hiperplano de separação entre as classes. Estes padrões são denominados de vetores-suporte (VS) e representam os vetores mais informativos do conjunto de treinamento. Desta forma, os vetores-suporte são os únicos vetores a serem armazenados para posterior apresentação e avaliação de padrões não vistos na etapa de teste.

Um fator limitante ao uso do classificador SVM é a complexidade de resolução [18], uma vez que treinar uma Máquina de Vetores-Suporte exige a resolução de um problema de programação quadrática (QP) com restrições de desigualdade [19]. Outro fator importante para sua devida utilização, consiste na escolha adequada da função de kernel e, consequentemente, da escolha dos parâmetros associados a mesma. Esta função é responsável por elevar a dimensão dos dados utilizados para que seja possível separá-los através de um hiperplano ¹. Diversas funções de kernel encontram-se descritas na literatura. Uma função comumente utilizada é a RBF (*Radial Basis Function*), dada por $K(\mathbf{x}, \mathbf{x}_i) = \exp(-\|\mathbf{x} - \mathbf{x}_i\|^2 / \sigma^2)$.

A formulação do problema primal de uma SVM é dada pela minimização da seguinte função custo:

$$\min J_P(\mathbf{w}, b, \xi) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^N \xi_i, \quad (17)$$

¹ Tal característica está associada a qualquer classificador que utilize uma função de kernel.

sujeito a $\begin{cases} y_i \{ [\mathbf{w}^T \phi(\mathbf{x}_i)] + b \} \geq 1 - \xi_i, & i = 1, \dots, N, \\ \xi_i \geq 0, & i = 1, \dots, N. \end{cases}$

em que a constante C representa uma regularização entre os dois termos da função $J_P(\mathbf{w}, b, \xi_i)$, sendo o primeiro termo $(\frac{1}{2} \mathbf{w}^T \mathbf{w})$ responsável pela maximização da margem e o segundo termo $(C \sum_{i=1}^N \xi_i)$ responsável por minimizar os erros permitidos.

Problemas dessa natureza que apresentam uma função custo convexa e com restrições lineares [8], podem ser solucionados através da aplicação do *método dos multiplicadores de Lagrange*. Este método agrega as restrições à função custo, que associadas a variáveis auxiliares, transformam o *problema primal* em um novo. Dessa forma, a seguinte função Lagrangeana $L(\mathbf{w}, b, \xi, \alpha, \beta)$ é obtida:

$$L(\mathbf{w}, b, \xi, \alpha, \beta) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \alpha_i \{ y_i [\mathbf{w}^T \phi(\mathbf{x}_i) + b] - 1 + \xi_i \} - \sum_{i=1}^N \beta_i \xi_i, \quad (18)$$

em que os conjuntos de multiplicadores de Lagrange $\{\alpha_i\}_{i=1}^N$ e $\{\beta_i\}_{i=1}^N$ possuem valores positivos.

A partir da solução do problema apresentado na Eq. 18, o problema dual a ser otimizado para o classificador SVM é expresso por:

$$\max L(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j, \quad (19)$$

$$\text{sujeito a } \begin{cases} \sum_{i=1}^N \alpha_i y_i = 0, \\ 0 \leq \alpha_i \leq C, & i = 1, \dots, N. \end{cases}$$

O processo de tomada de decisão é realizado com base na seguinte função

$$f(\mathbf{x}) = \text{sign} \left(\sum_{i=1}^N \alpha_i^o y_i \mathbf{x}^T \mathbf{x}_i + b_o \right), \quad (20)$$

em que valores positivos são associados a padrões pertencentes à classe $y_i = +1$ (representado, por exemplo, a existência de defeito em um isolador), enquanto valores negativos são associados a padrões pertencentes à classe $y_i = -1$ (representado, por exemplo, um isolador normal).

Como dito anteriormente, uma SVM também tem a capacidade de resolver problemas não-lineares pelo mapeamento do espaço de entrada para um de altíssima dimensão, isto é feito no problema dual pela substituição do produto interno $\mathbf{x}_i^T \mathbf{x}_j$ por um função de *kernel* $K(\mathbf{x}_i, \mathbf{x}_j)$ tanto na Equação 19 quanto na Equação 20.

III. SIMULAÇÕES COMPUTACIONAIS

A. Módulo de Aquisição

Para a captação dos dados utilizados na montagem da base de dados de treinamento das redes neurais, foram utilizados cinco isoladores poliméricos de linha de 69kV nas seguintes

condições de deterioração a saber: isoladores secos e isoladores molhados com água salgada. Todos os isoladores foram submetidos à tensão nominal.

Os isoladores foram elevados por um cabo conectado ao terra pela parte superior e energizados pela fase, parte inferior do isolador, como se pode ver na Figura 1. Para minimizar interferências, um cano metálico foi usado para envolver o cabo que liga o transformador ao isolador.



Figura 1: Método de Fixação do Isolador para Ensaio

Por meio do equipamento SDT-200, é possível identificar se há a presença de efeito corona nos isoladores enquanto submetidos à tensão elétrica. Assim, foram captados sinais com e sem a presença de efeito corona. Se faz necessário rotular o sinal coletado porque, como já mencionado, o método de aprendizagem das redes utilizadas é supervisionado.

Para captação do sinal foi desenvolvido um protótipo de aquisição de dados baseado no kit de desenvolvimento STM32F407 DISCOVERY. A taxa de amostragem utilizada foi de 225 mil amostras por segundo, onde cada captura teve duração de 22 ms e 5000 amostras. Cada sinal lido (contendo as 5000 amostras) foi armazenado em uma memória Flash para posterior análise.

Desta forma, obteve-se uma base de dados composta por 179 padrões com 12 atributos (veja mais detalhes na subseção abaixo), dentre os quais 90 padrões são rotulados como presença de efeito corona e 89 são rotulados como ausência de efeito corona.

B. Pré-Processamento

Cada sinal $s(t)$ capturado deve ser utilizado para descrever um padrão para a rede neural e cada amostra deste sinal descreve o valor para um atributo (característica) do padrão. É nesta etapa de pré-processamento que ocorre a extração de atributos mais significativos do sinal original e consiste nas seguintes etapas:

- 1) normalização da amplitude do sinal $s(t)$, através da divisão de cada atributo por 4096 (uma vez que a resolução do conversor analógico-digital é de 12 bits);

- 2) retirada do valor médio do sinal, como segue: $s(t) = s(t) - \text{mean}(s(t))$;
- 3) identificação de picos, em que se considera pico qualquer atributo cujo valor absoluto for 3 vezes maior que a variância do sinal $s(t)$;
- 4) definição de um vetor $\mathbf{p}$ contendo os dez maiores picos de $s(t)$;
- 5) cálculo da média $\bar{p}$ e do desvio padrão σ_p do vetor $\mathbf{p}$;
- 6) montagem do padrão que representa o sinal com 5000 amostras no seguinte formato:

$$\langle p_1 \rangle \langle p_2 \rangle \dots \langle p_{10} \rangle \langle \bar{p} \rangle \langle \sigma_p \rangle \langle \text{classe} \rangle$$

em que $\langle \text{classe} \rangle$ é o rótulo do sinal verificado pelo instrumento SDT-200 (com corona ou sem corona).

A Figura 2 apresenta o resultado parcial da etapa de pré-processamento realizada em sinal sem a presença de efeito corona. De maneira similar, o mesmo pode ser visto agora em

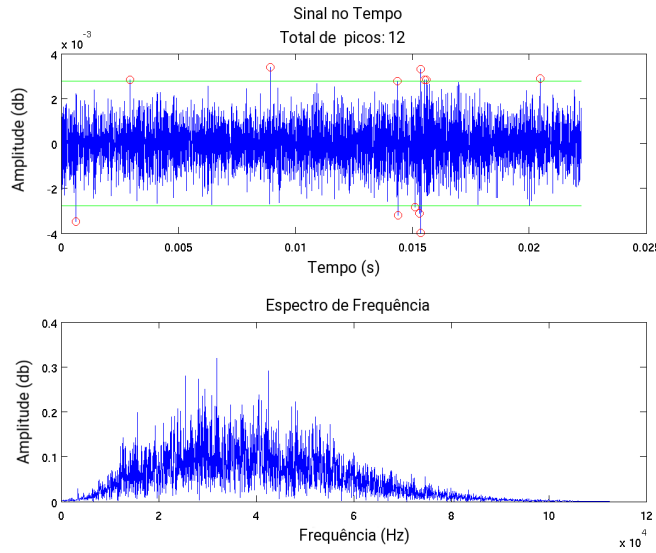


Figura 2: Resultado parcial do pré-processamento em um sinal sem efeito corona.

um sinal com efeito corona na Figura 3.

C. Normalização dos Dados

As redes utilizadas foram implementadas utilizando o ambiente *Matlab* e sua respectiva linguagem de programação nativa. Nas simulações computacionais realizadas para treinamento/teste das redes neurais artificiais, o primeiro procedimento a ser realizado foi a normalização do conjunto de dados em função da sua média e desvio padrão de cada um dos atributos. Assim, um padrão normalizado pode ser obtido por

$$x_{i_{norm}} = \frac{x_i - \bar{x}_i}{\sigma_i}, \quad (21)$$

em que x_i é o i -ésimo atributo de um padrão, bem como $\bar{x}_i$ e σ_i são a média e o desvio padrão também para o i -ésimo atributo.

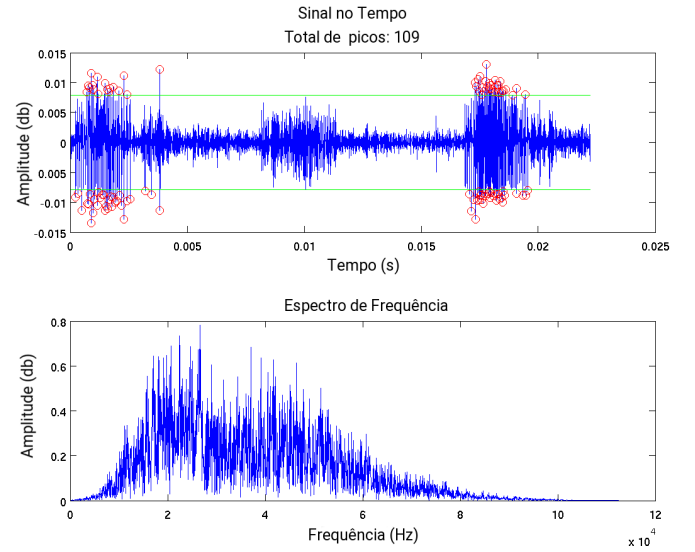


Figura 3: Resultado parcial do pré-processamento em um sinal com efeito corona.

D. Metodologia para Treinamento/Teste

Em cada realização, o conjunto total de dados foi dividido em dois, um para treinamento e outro para teste. Os padrões do conjunto de treinamento são utilizados inicialmente para obtenção do melhor conjunto de parâmetros das redes neurais através de um método de busca em grade (*gridsearch*) combinado com um método de validação cruzada (*cross validation*) [20]. Em seguida, esse melhor conjunto de parâmetros obtido é usado para treinar a rede neural, mas agora utilizando todo o conjunto de treinamento.

Em termos de arquitetura, a rede MLP possui apenas uma camada oculta e uma de saída. A rede MLP foi treinada por 200 épocas. A taxa de aprendizagem η e o número de neurônios na camada oculta foram definidos, como se sabe, a partir da busca em grade anteriormente mencionada. O classificador SVM utiliza kernel RBF e os valores dos parâmetros C (termo de regularização) e σ (necessário pelo tipo de kernel utilizado) também foram definidos a partir da busca em grade.

Para que as redes sejam avaliadas, em cada realização, em termos de capacidade de generalização é necessário que padrões ainda não apresentados sejam apresentados a rede já treinada, como descrito na subseção anterior. Assim, os padrões de teste reservados com essa finalidade são apresentados à rede. Veja que os mesmos não foram utilizados em nenhum momento para o treinamento da rede.

E. Resultados Obtidos

A fim de comparar o desempenho dos classificadores MLP e SVM, apresenta-se na Tabela I a taxa média de acerto (acurácia) para o conjunto de teste juntamente com seu desvio padrão para 30 realizações em função da variação do tamanho do conjunto de treinamento ($\#CT$).

Tabela I: Resultados para as redes MLP e SVM em função da variação do tamanho do conjunto de treinamento.

Modelo	# CT	Taxa de Acerto	Desvio Padrão
MLP	20%	80,75	$\pm 10,7$
	50%	82,58	$\pm 10,6$
	80%	84,81	$\pm 8,6$
SVM	20%	85,40	$\pm 4,8$
	50%	88,39	$\pm 2,9$
	80%	88,33	$\pm 6,4$

Ao analisar os dados da Tabela I, pode-se concluir que o desempenho dos classificadores avaliados apresentam valores de acurácia superiores a 80%, independente do tamanho do conjunto de treinamento utilizado.

Na Figura 4 são apresentados os valores da taxa média de acerto para 10%, 20%, 30%, $\dots$, 80% do tamanho do conjunto de treinamento.

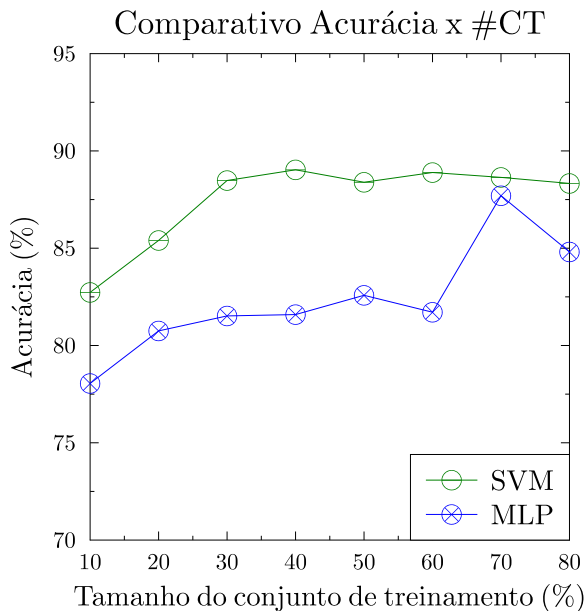


Figura 4: Acurácia versus tamanho do conjunto de treinamento.

Nota-se na Figura 4 que o desempenho obtido pelo classificador SVM apresenta-se melhor do que o desempenho obtido pela rede MLP, independentemente do percentual usado para treinamento.

IV. CONCLUSÃO

Neste trabalho foi proposta a utilização de redes neurais artificiais como ferramenta de diagnóstico de efeito corona em isoladores poliméricos de alta tensão. Com base nos resultados obtidos, conclui-se que ambos os modelos MLP e SVM utilizados apresentam alta capacidade de generalização e são capazes de inferir a presença ou ausência de efeito corona a partir de ruídos ultrassônicos provenientes dos isoladores.

Nota-se ainda que os modelos avaliados apresentam acurácias equiparáveis, podendo chegar até cerca de 89% para o classificador SVM e 87% para o classificador MLP.

REFERÊNCIAS

- [1] C. L. Borges. Uma abordagem alternativa para inspeção preditiva não invasiva em linha aérea de distribuição de média tensão na área rural. Doutorado, Universidade Federal de Itajubá (UNIFEI), 2012. Programa de pós-graduação em Engenharia Elétrica, UNIFEI, Minas Gerais.
- [2] L.P. Alessi and S.K. Ogawa. Estudo comparativo entre as técnicas aplicadas à inspeção instrumentalizada de redes de distribuição de energia elétrica. Monografia, Universidade Federal do Paraná (UFPR), 2010. Departamento de Engenharia Elétrica, UFPR, Curitiba.
- [3] M.G.G. Neri, E.G. Costa, R.W.S. Garcia, and O.L.S. Paiva. Avaliação de técnicas de monitoramento de isoladores poliméricos. In *6º Encuentro de Potencia, Instrumentación y Medidas (EPIM, 2005)*. Montevideo, 2005.
- [4] W.C. Godoi, R.R. Silva, V. Swinka-Filho, and T. Loddí. Detecção de defeitos em isoladores poliméricos por meio da radiografia digital e reconhecimento de padrões. In *Espaço Energia: Revista técnicocientífica da área de energia*. 2 edition, 2005.
- [5] C.R. Gomes. Localização de falhas em linhas de transmissão utilizando decomposição harmônica e redes neurais artificiais. Mestrado, Universidade Federal do Pará (UFPA), 2006. Programa de Pós-Graduação em Engenharia Elétrica, UFPA, Pará.
- [6] J.M.B. Bezerra, R.R.B. Aquino, Oliveira J.B., O. Nóbrega, T.M.A. Silveira, E.G. Costa, M.G.G. Néri, T.V. Ferreira, J.L.P. Dantas, and P.L. Mendonça. Avaliação de sensor de ultra-som como técnica preditiva na manutenção de subestações e linhas de transmissão e distribuição. In *V Congresso de Inovação Tecnológica em Energia Elétrica (V CITENEL)*. Belém, Pará, 2009.
- [7] R.M. Capelini. Localização de cadeias de isoladores defeituosos em linhas de transmissão através de redes neurais. Mestrado, Universidade Federal de Itajubá (UNIFEI), 2011. Programa de Pós-Graduação em Engenharia Elétrica, UNIFEI, Minas Gerais.
- [8] S. Haykin. *Redes Neurais: Princípios e práticas*. Bookman, Porto Alegre, 2 edition, 2001.
- [9] W.S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biology*, 5:115–133, 1943.
- [10] C.V. Ribeiro, R.R. Goldschmidt, and N.R. Choren. Métodos para previsão de séries temporais e suas tendências de desenvolvimento. Monografia, Instituto Militar de Engenharia (IME), 2009. Sistemas e Computação, IME, Rio de Janeiro.
- [11] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.
- [12] V. Vapnik and A. Lerner. Pattern recognition using generalized portrait method. *Automation and Remote Control*, 24, 1963.
- [13] B.E. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 144–152, New York, NY, 1992. ACM Press.
- [14] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, Inc., New York, NY, USA, 1995.
- [15] Robert Burbidge and Bernard Buxton. An introduction to support vector machines for data mining. In *Keynote Papers, Young Or12, University of Nottingham, Operational Research Society, Operational Research Society*, pages 3–15, Cambridge, UK, 2001. Press.
- [16] C. Cortes and V. Vapnik. Support-vector networks. *Mach. Learn.*, 20:273–297, 1995.
- [17] S.R. Gunn. Support vector machines for classification and regression. Technical report, Faculty of Engineering, Science and Mathematics. School of Electronics and Computing Science, 1998.
- [18] B.P.R. Carvalho. Novas estratégias para detecção automática de vetores de suporte em least squares support vector machines. Mestrado, Universidade Federal de Minas Gerais (UFMG), 2005.
- [19] E. Osuna, R. Freund, and F. Girosi. An improved training algorithm for support vector machines. In *Neural Networks for Signal Processing [1997] VII. Proceedings of the 1997 IEEE Workshop*, pages 276–285, Amelia Island, FL, Sep 1997. IEEE.
- [20] C.W. Hsu, C.C. Chang, and C.J. Lin. A practical guide to support vector classification. Technical report, Department of Computer Science, National Taiwan University, 2003.