

A decision support system based on Machine Learning and Fuzzy Logic Techniques for electrical fault detection and classification

Davi Carvalho, José Melo, Eraylson G. Silva, Paulo S. G. de Mattos Neto

¹Centro de Informática - Universidade Federal de Pernambuco (CIn/UFPE)
Caixa Postal 15.064 – 91.501-970 – Recife – Pernambuco – Brasil

{dsc6, jfvm, egs, psgmn}@cin.ufpe.br

Abstract. *The detection and classification of faults in electrical systems are crucial for ensuring uninterrupted power supply. This paper proposes a decision support system that combines Machine Learning and Fuzzy Logic techniques to detect, classify, and diagnose faults in electrical systems. The system comprises Classification Module and Rules Module. The first one employs a Decision Tree to generate a set of rules and classify the current system status. The second one allows the operator to manipulate this knowledge basis to generate more intelligible and interpretable rules using Fuzzy Logic. The results using a well-known data set show the proposal overperforms other classification models and can generate more straightforward and interpretable rules.*

Resumo. *A detecção e classificação de falhas em sistemas elétricos são cruciais para garantir fornecimento de energia. Este artigo propõe um sistema de apoio à decisão que combina técnicas de Aprendizado de Máquina e Lógica Fuzzy para detectar, classificar e diagnosticar falhas elétricas. O sistema é composto pelos Módulos de Classificação e de Regras. O primeiro utiliza uma Árvore de Decisão para gerar um conjunto de regras e classificar o estado atual do sistema. O segundo permite ao operador manipular essa base de conhecimento para gerar regras mais inteligíveis e interpretáveis usando Lógica Fuzzy. Os resultados com um conjunto de dados bem conhecido mostram que a proposta supera outros classificadores e pode gerar regras mais simples e interpretáveis.*

1. Introdução

Os atuais sistemas de energia elétrica são compostos por diversos componentes [Goni et al. 2023]. Esses componentes elétricos são dinâmicos e estão propensos a perturbações ou falhas, que podem ocorrer por causa de algum problema na rede (e.g. algum curto-circuito), ou por fatores externos, como questões ambientais [Janarthanam et al. 2022]. O emprego de redes de usinas de geração elétrica de alta capacidade trabalhando de forma sincronizada necessita que todo o sistema de energia opere em condições estáveis. Nesse sentido, é de extrema importância que as falhas sejam detectadas para que os procedimentos de proteção sejam executados no menor tempo possível, visando a proteção dos equipamentos e evitando maiores danos a rede [Jamil et al. 2015].

Dentre os problemas que podem ocorrer na rede, as falhas nas linhas de transmissão do sistema devem ser detectadas e classificadas para que sejam solucionadas o

mais rápido possível [Jamil et al. 2015]. Assim, é desejável que um sistema de detecção de falhas seja eficiente e confiável para auxiliar na decisão do operador.

Com o objetivo de desenvolver sistemas acurados e eficientes de detecção a falhas elétricas, técnicas de Aprendizado de Máquina (AM) têm sido empregadas nessa tarefa [Goni et al. 2023, Janarthanam et al. 2022, Jamil et al. 2015, Upendar et al. 2008, Fahim et al. 2020]. Dentre essas técnicas, as Redes Neurais Artificiais (RNAs) têm sido largamente empregadas por ser uma família de modelos treináveis, flexíveis, não-lineares, que tem alcançado resultados de destaque em diversas tarefas que envolvem reconhecimento de padrões, tais como classificação, regressão e previsão de séries temporais.

Mukherjee *et al.* [Mukherjee et al. 2020] usaram uma abordagem combinada de Análise de Componentes Principais e RNAs probabilísticas para analisar e classificar falhas em uma linha de transmissão aérea trifásica de 150 km. A abordagem proposta obteve 100% de precisão na classificação de falhas elétricas, mesmo considerando incertezas práticas como ruído do sistema de potência e resistência de falta variável [Mukherjee et al. 2020].

Omar *et al.* [Omar et al. 2020] propuseram a rede *Long Short-Term Memory* (LSTM) como ferramenta para classificar diferentes tipos de falhas. Neste estudo, foi modelado uma linha de transmissão com 400 kV e 100 km de distância com dez classes de sinais de falha. Os sinais de falha foram pré-processados extraíndo os sinais de corrente pós-falha. Em seguida, esses sinais foram dados como entrada para a rede LSTM com o objetivo de classificar 10 tipos de falhas. Nos três conjuntos de dados testados, o algoritmo obteve 100%, 99.77% e 99.55%, respectivamente.

Samantaray *et al.* [Samantaray et al. 2007] empregaram um modelo de Máquina de Vetor de Suporte (*Support Vector Machine* - SVM) para detecção de falhas em uma ou múltiplas fases da linha de transmissão. Uma vez que as fases defeituosas são identificadas, a localização da falha foi estimada utilizando uma RNA de função de base radial (*Radial Basis Function Neural Network* - RBFNN) com algoritmo recursivo de mínimos quadrados. A combinação de SVM e RBFNN foi testada em uma variedade de condições operacionais, fornecendo a localização e a classificação das falhas. As taxas de acerto alcançadas ficaram acima de 97% para faltas com grandes variações nas condições de operação.

Fahim *et al.* [Fahim et al. 2020] propuseram um novo modelo de rede neural convolucional de auto atenção (*Self-Attention Convolutional Neural Network* - SAT-CNN) para detecção e classificação de falhas em linhas de transmissão com modelo de extração de características baseado em imagens de séries temporais. Para garantir a imunidade ao ruído, a transformada discreta de wavelet foi usada para remover o ruído dos sinais de tensão e corrente com falha. O modelo SAT-CNN foi testado variando os sinais de entrada (tensão, corrente e sinal) sob as várias frequências de amostragem, obtendo 91,26% de taxa de acerto.

Janarthanam *et al.* [Janarthanam et al. 2022] testaram diferentes técnicas de AM para detecção e classificação de seis tipos de falhas em linhas de transmissão: SVM, Árvore de Decisão, *Random Forest Classifier* e *K-Nearest Neighbors* (KNN). O SVM obteve o melhor desempenho entre os modelos, com 0,9969 de precisão na detecção de falhas.

Goni *et al.* [Goni et al. 2023] propuseram uma nova abordagem para detecção e classificação de falhas. A abordagem foi aplicada em duas linhas de transmissão (LT-1 e LT-2) simuladas usando MATLAB. Nessas bases dados de tensão e corrente foram geradas para simular dez tipos de falhas. A abordagem utilizou uma Extreme Learning Machine (ELM) como classificador. A proposta alcançou uma taxa de acerto de 99,18% e 99,09% para as LT-1 e LT-2, respectivamente. As precisões de detecção de falhas foi de 99,53% e 99,60% foram alcançadas para os LT-1 e LT-2. O modelo ELM apresentou melhor desempenho e menor complexidade computacional que outras técnicas de AM.

Jamil *et al.* [Jamil et al. 2015] aplicaram uma RNA do tipo *Multilayer Perceptron* (MLP) para detecção e classificação de falhas em um sistema trifásico de linhas de transmissão. O método desenvolvido utiliza tensões e correntes trifásicas como entradas. Os resultados alcançados mostram que para testes de falha do tipo linha-terra, a MLP obteve uma taxa de acerto de 78,1%.

Apesar dos resultados de destaque em termos de desempenho, as RNAs são modelos *black box*, provendo para o operador apenas o estado atual do sistema. Assim, o operador do sistema de energia toma a decisão baseado apenas na saída do modelo de classificação sem ter acesso a qualquer explicação ou regra que possa ter gerado a falha. Contudo, o entendimento da causa do erro pode possibilitar a melhoria do sistema, além de prover *insights* que podem ser utilizados como insumo de diagnósticos preditivos, visando evitar futuras falhas. Além disso, as regras podem aumentar o conhecimento do operador do sistema também em estados de não falha.

Esse trabalho propõe um sistema de apoio à decisão baseado em técnicas de AM para detecção e diagnóstico de falhas elétricas. A proposta é composta de duas fases: primeiramente uma árvore de decisão é utilizada para classificar o estado atual do sistema baseado em grandezas elétricas, após isso um modelo de Lógica *Fuzzy* [Toliat et al. 1996] é empregado para tornar as regras interpretáveis baseado no conhecimento do operador. Assim, a saída da abordagem consiste no estado atual do sistema e no conjunto de regras inteligíveis que geraram essa classificação. O sistema é avaliado em uma base de dados pública geradas a partir de um sistema de energia simulado no MATLAB [MathWorks 2022]. O sistema é comparado com outros modelos de AM da literatura utilizando métricas bem conhecidas na tarefa de classificação. Os resultados demonstram que o sistema proposto obtém desempenho superior que outros modelos da literatura e que é capaz de gerar regras mais simples, inteligíveis e interpretáveis, aumentando as chances do operador do sistema tomar a melhor decisão de forma mais rápida e eficiente.

2. Sistema Proposto

A estrutural geral do sistema proposto pode ser vista na Figura 1. O sistema é dividido em dois módulos: Módulo de Classificação e Módulo de Regras. O Módulo de Classificação é responsável pela geração de regras e das respectivas classes. O Módulo de Regras é utilizado pelo especialista para tornar as regras mais inteligíveis e, ocasionalmente, criar novas regras. Nas Seções 2.1 e 2.2 os dois módulos que compõem o sistema proposto serão detalhados.

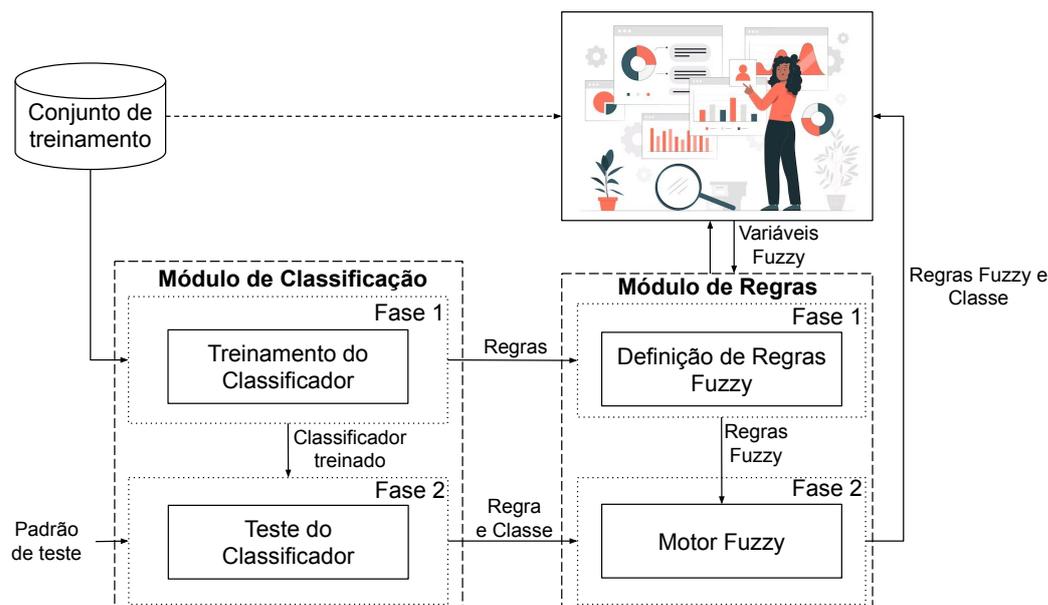


Figura 1. O sistema proposto é composto pelos Módulos de Classificação e de Regras. Cada módulo possui as Fases 1 e 2.

2.1. Módulo de Classificação

O Módulo de Classificação é composto de duas fases: treinamento (Fase 1) e teste (Fase 2). Na fase de treinamento, o classificador recebe como entrada o conjunto de treinamento. A saída dessa fase é o classificador treinado e as regras correspondentes. Qualquer classificador *white-box* pode ser escolhido para ser utilizado nesse módulo. A escolha desse tipo de modelo é importante porque o objetivo da abordagem é gerar o diagnóstico do sistema energia, aumentando a consciência situacional do especialista. Nesse sentido, as regras relacionam as variáveis elétricas com o estado atual do sistema. Entretanto, é crucial que as regras possam ser entendidas e interpretadas com facilidade pelo especialista do sistema de energia, possibilitando que uma tomada de decisão seja realizada o mais rápido possível, evitando que outros componentes do sistema sejam afetados.

As regras (anteriores com a respectiva classe) geradas na fase de treinamento são inseridas como entrada no Módulo de Regras. Essas regras podem ser posteriormente visualizadas, manipuladas e editadas pelo operador no Módulo de Regras.

Na fase de teste do Módulo de Classificação, o classificador treinado recebe como entrada um padrão de teste. O padrão de teste é um vetor composto por um novo conjunto de variáveis do sistema elétrico. Para cada novo padrão o Módulo de Classificação verifica a regra acionada no classificador e a respectiva classe. Para cada padrão de teste, a saída do Módulo de Classificação é a regra acionada e a classe correspondente.

2.2. Módulo de Regras

O Módulo de Regras é utilizado pelo especialista para tornar as regras mais interpretáveis e para a criação de novas regras. Esse módulo é utilizado pelo especialista para interagir com o sistema proposto. Nesse sentido, o operador do sistema pode ter acesso aos dados de treinamento se achar necessário para a criação das variáveis *fuzzy*.

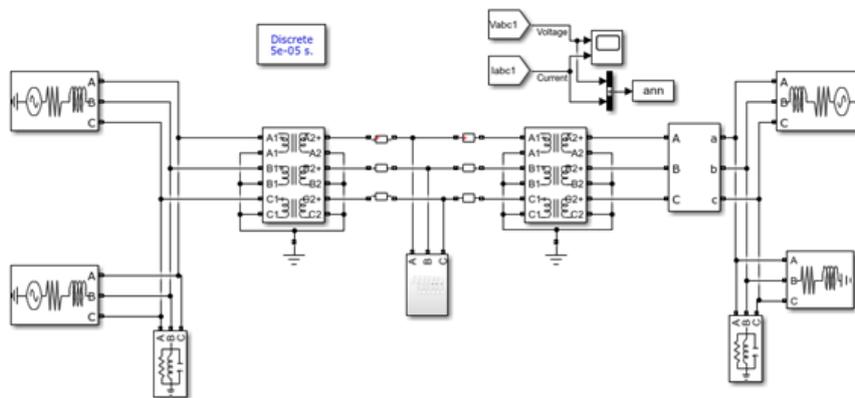


Figura 2. Sistema de transmissão de energia simulado com quatro geradores e dois transformadores. Fonte: [kaggle.com/datasets/esathyaprakash/electrical-fault-detection-and-classification](https://www.kaggle.com/datasets/esathyaprakash/electrical-fault-detection-and-classification)

Esse módulo é dividido em duas fases: Definição de Regras *Fuzzy* (Fase 1) e Motor *Fuzzy* (Fase 2). Na primeira fase, o Módulo de Regras recebe como entrada as expressões condicionais geradas pelo Módulo de Classificação. Essas regras são do tipo IF-THEN-ELSE e relacionam as características elétricas com a situação atual do sistema (classe). Com as regras em mãos, o operador pode analisar as regras e criar variáveis *fuzzy* com o objetivo de aumentar a interpretabilidade e explicabilidade das regras. Essa fase é importante porque permite que o operador tenha uma maior consciência do estado atual do sistema elétrico e como as variáveis estão relacionadas com possíveis falhas.

Na Fase 2, o Motor *Fuzzy* recebe como entrada do Módulo de Classificação a regra ativada com a respectiva classe. O Motor *Fuzzy* será responsável por traduzir a regra utilizando as variáveis *fuzzy* criadas pelo operador. Após isso, o Módulo de Regras envia a regra *fuzzy* com a respectiva classe para o operador tomar uma decisão se necessário. Para cada conjunto de variáveis do sistema, a proposta gera regra e a classe correspondente.

3. Protocolo dos Experimentos e Análise dos Resultados

O sistema proposto foi analisado utilizando uma base de dados pública¹. A base *Electrical Fault Detection and Classification* (Detecção e Classificação de Falhas Elétricas) é composta por uma coleção de estados que englobam valores de corrente e tensão em linhas de transmissão para diferentes condições do sistema elétrico. Na Figura 2 pode ser visto o sistema de energia foi simulado no Matlab e é composto por 4 geradores de 11×10^3 V. Cada par de geradores é localizado nas extremidades da linha de transmissão. Os transformadores estão localizados no centro com o objetivo de simular diferentes falhas na linha de transmissão.

O circuito foi simulado em condições normais e em várias condições de falha. A base de dados é composta por 7.861 instâncias. Cada instância tem 6 atributos: I_a , I_b , I_c , V_a , V_b , V_c ; e pode ter uma de 6 classes. Os valores das variáveis I_a , I_b e I_c correspondem às correntes das linhas nas fases A, B e C, respectivamente, enquanto as variáveis V_a , V_b , V_c são as tensões nas linhas das A, B e C, respectivamente. Utilizando essas variáveis

¹<https://www.kaggle.com/datasets/esathyaprakash/electrical-fault-detection-and-classification>

Tabela 1. Métricas para avaliação da classificação. Na coluna Valor Desejável, o símbolo ↑ significa que quanto maior o valor da métrica melhor é a desempenho na classificação.

Métrica	Variável	Equação	Limites	Valor Desejável
Acurácia	-	$\frac{VP+VN}{VP+VN+FP+FN}$	[0, 1]	↑
Precisão	P	$\frac{VP}{VP+FP}$	[0, 1]	↑
Sensibilidade	S	$\frac{VP}{VP+FN}$	[0, 1]	↑
<i>F1-score</i>	-	$\frac{2.P.S}{P+S}$	[0, 1]	↑

como entrada, o modelo de AM será treinado para classificar o estado do sistema entre as seguintes classes:

- Classe 0: não existe a ocorrência de falhas;
- Classe 1: ocorrência de falha entre a Fase A e Fase B;
- Classe 2: ocorrência de falha em todas as fases;
- Classe 3: ocorrência de falha e está acontecendo entre a Fase A e a Fase G;
- Classe 4: ocorrência de falha entre a Fase A, B e G;
- Classe 5: ocorrência de falha trifásica simétrica.

Para o Módulo de Classificação uma Árvore de Decisão [Toraih et al. 2020] foi selecionada como classificador porque tem alcançado resultados de destaque em termos de acurácia e é um modelo com alto nível de interpretabilidade [Sagi and Rokach 2021]. Com o objetivo de avaliar o desempenho do Módulo de Classificação do sistema proposto, outros classificadores baseados em árvores foram selecionados para comparação: XGBoost [Liew et al. 2021], LightGBM [Pan et al. 2023] e CatBoost [Al-Sarem et al. 2021]. Esses modelos foram escolhidos porque eles são capazes de gerar regras que são construídas na fase de treinamento para realizar a classificação, caracterizando-os como modelos *white-box* [Wilfling et al. 2022, Gadzinski and Castello 2022].

Com o objetivo de avaliar o Módulo de Classificação em relação a outros modelos de classificação da literatura, quatro métricas bem conhecidas foram escolhidas: Acurácia, Precisão, Sensibilidade e *F1-Score*. Nas métricas Acurácia, Precisão e Sensibilidade, VP, VN, FP e FN correspondem ao verdadeiro positivo, verdadeiro negativo, falso positivo e falso negativo, respectivamente. Na Tabela 1 as métricas consideradas podem ser visualizadas. Em todas, quanto mais alto o valor da métrica, melhor é o desempenho do modelo.

Tabela 2. Parâmetros utilizados pelo sistema proposto e modelos da literatura avaliados nos experimentos.

Classificador	Hiperparâmetro	Valores
Árvore de Decisão	critério	gini e entropy
	profundidade máxima	2, 3, 4, 5, 6, 7, 8, 9 e 10
XGBoost	taxa de aprendizagem	0,1, 0,01 e 0,001
	profundidade máxima	2, 3, 4, 5, 6, 7, 8, 9 e 10
	quantidade de estimadores	2, 4, 8 e 10
LightGBM	taxa de aprendizagem	0,1, 0,01 e 0,001
	profundidade máxima	2, 3, 4, 5, 6, 7, 8, 9 e 10
	quantidade de estimadores	2, 4, 8 e 10
CatBoost	taxa de aprendizagem	0,1, 0,01 e 0,001
	profundidade máxima	2, 3, 4, 5, 6, 7, 8, 9 e 10
	quantidade de estimadores	2, 4, 8 e 10

A base de dados foi dividida em dois conjuntos: 80% das instâncias para a amostra de treinamento e 20% para a amostra de teste. As instâncias foram selecionadas de forma estratificada com o objetivo de gerar conjuntos de treinamento e teste com a mesma proporção de instâncias de cada classe. Na etapa de treinamento dos modelos, os valores para os hiperparâmetros (Tabela 2) foram selecionados utilizando a abordagem de *Grid-search*, visando maximizar a Acurácia média na validação cruzada utilizando 5 *folds*.

3.1. Resultados do Módulo de Classificação

A Tabela 3 apresenta os resultados no conjunto de teste do sistema proposto e dos demais modelos de classificação da literatura. Para os casos onde não existe a ocorrência de falhas (Classe 0), os modelos Árvore de Decisão e LightGBM apresentaram desempenho máximo em todas as métricas avaliadas. Para os casos em que o estado do sistema apresenta a ocorrência de falhas entre Fase A e Fase B (Classe 1), os modelos Árvore de Decisão e LightGBM apresentaram os melhores resultados, com destaque para o LightGBM que apresentou um *F1-score* de 1.

Para as instâncias que apresentam falhas em todas as fases (Classe 2), é possível observar que os modelos apresentaram baixo desempenho em todas as métricas. Dentre os classificadores avaliados, apenas a Árvore de Decisão alcançou um *F1-score* acima de 50%, mostrando superioridade nessa classe.

Para as falhas que ocorrem entre a Fase A e a Fase G (Classe 3), o modelo com maior desempenho novamente foi a Árvore de Decisão. Nessa classe, o segundo melhor desempenho foi alcançado pelo LightGBM, obtendo os segundos melhores valores para Precisão e *F1-score*.

Na classe de falhas que apresenta problemas entre as Fases A, B e G (Classe 4), os modelos Árvore de Decisão e LightGBM apresentaram os melhores resultados. Ambos apresentaram uma Precisão de 1.00 e um *F1-score* de 0.98.

Na Classe 5, em que as instâncias apresentam falhas trifásicas simétricas, os modelos apresentaram baixo desempenho. Por exemplo, os dois melhores desempenhos foram obtidos pelo XGBoost e Árvore de Decisão, respectivamente. A Árvore de Decisão

Tabela 3. Resultados dos modelos de classificação no conjunto de teste.

Modelo	Métricas			
		Precisão	Sensibilidade	F1-score
Árvore de Decisão	Classe 0	1.00	1.00	1.00
	Classe 1	1.00	0.99	0.99
	Classe 2	0.53	0.65	0.58
	Classe 3	0.98	1.00	0.99
	Classe 4	1.00	0.95	0.98
	Classe 5	0.55	0.45	0.49
	Acurácia	0.88		
XGBoost	Classe 0	0.97	1.00	0.98
	Classe 1	0.99	0.96	0.98
	Classe 2	0.43	0.38	0.40
	Classe 3	0.95	0.94	0.94
	Classe 4	0.97	0.93	0.95
	Classe 5	0.47	0.54	0.50
	Acurácia	0.83		
LightGBM	Classe 0	1.00	1.00	1.00
	Classe 1	0.99	1.00	1.00
	Classe 2	0.40	0.43	0.41
	Classe 3	0.97	1.00	0.98
	Classe 4	1.00	0.97	0.98
	Classe 5	0.42	0.39	0.40
	Acurácia	0.84		
Catboost	Classe 0	0.96	1.00	0.98
	Classe 1	0.92	0.96	0.94
	Classe 2	0.45	0.52	0.48
	Classe 3	0.89	0.95	0.92
	Classe 4	0.90	0.91	0.90
	Classe 5	0.39	0.24	0.29
	Acurácia	0.81		

alcançou o melhor valor de Precisão, enquanto o XGBoost obteve os maiores valores de Sensibilidade e *F1-score*.

Em relação a métrica Acurácia, o sistema proposto empregando a Árvore de Decisão alcançou o melhor resultado. O seu desempenho foi superior aos demais, alcançado uma diferença relevante em relação ao segundo melhor modelo, o LightGBM. Esse resultado mostra que o sistema proposto classificou corretamente um número maior de instâncias em comparação com os demais modelos da literatura.

As Figuras 3, 4, 5 e 6 apresentam a matriz de confusão obtida para cada classificador analisado. No eixo X pode ser vista a classe que foi predita pelo modelo e no eixo Y são exibidas a classe real das instâncias no conjunto de teste. Na diagonal principal é mostrado o número de instâncias corretamente classificadas para cada classe. Quanto maior esse valor, melhor o desempenho do modelo para uma determinada classe. Através das matrizes pode-se observar a relação de erros entre diferentes classes, gerando alguns

insights sobre possíveis semelhanças entre duas ou mais categorias.

Analisando as matrizes de confusão (Figuras 3, 4, 5 e 6) é possível notar que todos os modelos classificaram corretamente a mesma quantidade de instâncias para a Classe 0. Para as classes 1, 4 e 5 o LightGBM foi o modelo que classificou corretamente a maior quantidade de instâncias. Para as classes 2 e 3 a Árvore de Decisão foi o modelo com maior quantidade de instâncias classificadas corretamente.

É possível observar nas matrizes de confusão que as classes 2 e 5 apresentam a maior quantidade de instâncias classificadas incorretamente pelos modelos. Além disso, é evidente que, na maioria dos casos, o modelo tende a retornar a classe 5 para instâncias que pertencem à classe 2, e o mesmo ocorre de forma inversa. Esse comportamento entre essas classes, pode indicar que seja necessário o conhecimento do especialista para diferenciar entre a situação de falhas em todas as fases e a ocorrência de falha trifásica simétrica.

Figura 3. Matriz de confusão obtida através das classificações da Árvore de Decisão no conjunto de teste.

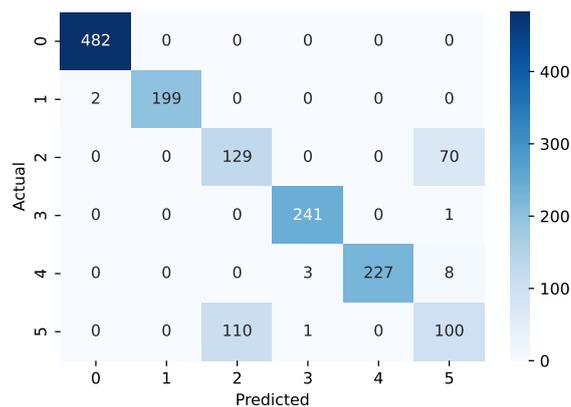


Figura 4. Matriz de confusão obtida através das classificações do XGBoost no conjunto de teste.

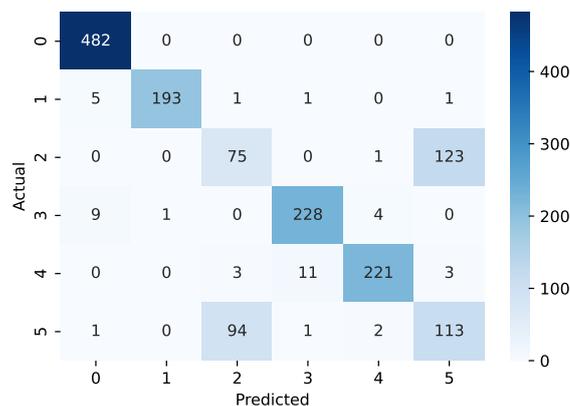


Figura 5. Matriz de confusão obtida através das classificações do LightGBM no conjunto de teste.

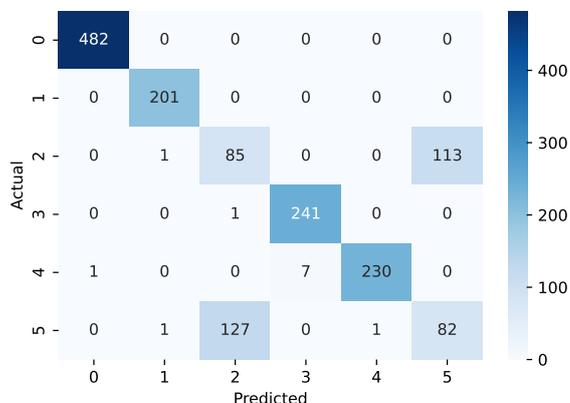
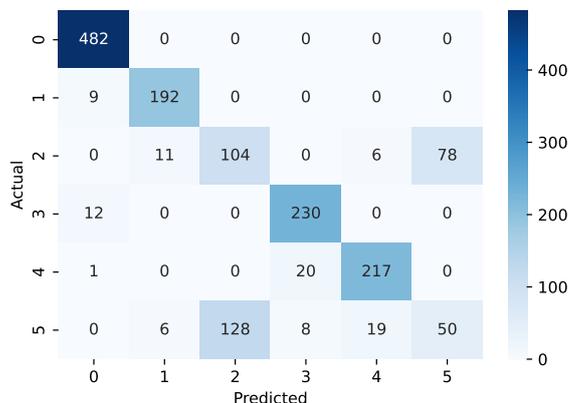


Figura 6. Matriz de confusão obtida através das classificações do CatBoost no conjunto de teste.



A partir da análise dos resultados obtidos na Tabela 3 e da comparação das matrizes de confusão, é possível notar que o Árvore de Decisão e o LightGBM foram os modelos que se destacaram em relação as métricas utilizadas. Apenas para duas classes (Classe 2 e Classe 5), esse modelos obtiveram um resultado satisfatório.

3.2. Análise do Módulo de Regras

Após gerar o modelo de classificação é possível analisar o conjunto de regras geradas pelo classificador. O Árvore de Decisão treinado apresenta 102 folhas. Esse número corresponde a quantidade de regras que podem ser extraídas dele. Esse conjunto de regras é a entrada para o Módulo de Regras.

A Figura 7 apresenta um exemplo de uma regra gerada pela Árvore de Decisão.

Figura 7. Exemplo de uma regra extraída do modelo Árvore de Decisão empregado pelo Módulo de Classificação.

```
[ 'Ic > 59.79086112976074' ,
  'Ia <= -61.533424377441406' ,
  'Ic <= 899.4051208496094' ,
  'Ia <= -882.9309387207031' ,
  'Classificação: 5' ]
```

Apesar da Árvore de Decisão ter apresentado melhor desempenho em comparação com os demais classificadores da literatura na maioria das classes e ser um modelo *white-box*, a estrutura de regras gerada dificulta a interpretação do especialista, como pode ser visto no exemplo da Figura 7. Além disso, na aplicação de monitoramento de plantas elétricas, muitas vezes é necessário ter uma forma do especialista interagir com o sistema, possibilitando que ele possa adicionar novas regras com o objetivo de aumentar o espaço de decisão visando detectar mais falhas. Essa funcionalidade é extremamente importante em casos em que o classificador não apresenta desempenho satisfatório.

Esses pontos corroboram com a necessidade de se ter um módulo capaz de tornar as regras geradas pelo classificador mais interpretáveis pelo especialista. Nesse sentido, é importante prover a flexibilidade para o especialista para que ele possa criar variáveis e regras mais inteligíveis.

No módulo de regras o usuário faz a leitura das regras com valores *crisp* que foram geradas pelo classificador e através do motor *fuzzy* as regras são transformadas em regras com variáveis contendo valores *fuzzy*. A Figura 8 apresenta o recorte de tela do sistema que apresenta um conjunto de regras utilizando as variáveis e seus respectivos valores *fuzzy*. Na primeira coluna os antecedentes podem ser vistos, ou seja, a parte da regra que contém as informações de entrada e as condições que precisam ser atendidas para ela ser disparada e o respectivo consequente (a classe apresentada na segunda coluna) seja gerado. Para a base utilizada, existem 6 consequentes possíveis que são as classes.

Inputs	Outputs
IF(Ia is Médio) and (Ib is Alto) and (Ic is Baixo) and (Va is Baixo) and (Vb is Médio) and (Vc is Alto)	THEN(Estado is Falha em todas as fases)
IF(Ia is Alto) and (Ib is Médio) and (Ic is Alto) and (Va is Alto) and (Vb is Baixa) and (Vc is Alto)	THEN(Estado is Sem falhas)
IF(Ia is Baixo) or (Ia is Muito Baixo) and (Ic is Médio) or (Ic is Alto)	THEN(Estado is Falha trifásica simétrica)

Figura 8. Exemplo de regras geradas no módulo de regras fuzzy. Recorte da tela do sistema utilizado pelo especialista.

Nesse módulo o especialista consegue fazer a seleção e interpretação das regras geradas pelo classificador, assim como observar quais regras estão associadas com um

determinado estado do sistema. No exemplo apresentado na Figura 8 podem ser vistas 3 regras: a primeira regra está associada ao estado em que o sistema apresenta falha em todas as fase; a segunda regra está associada ao estado em que o sistema não apresenta falha; e a terceira regra está associada ao estado em que o sistema apresenta uma falha trifásica simétrica.

A terceira regra apresentada na Figura 8 é o resultado da transformação realizada pelo Módulo de Regras na regra apresentada na Figura 7. Como pode-se observar, esse módulo é capaz de tornar a regra mais simples e inteligível para que o especialista consiga tomar as melhores decisões.

Além do conjunto de regras geradas pelo classificador, o especialista pode adicionar novas regras. Essa funcionalidade permite adicionar o conhecimento do especialista no sistema. Para a base de dados avaliada nesse trabalho, a adição de novas regras possibilita aumentar a acurácia nas classes 2 e 5, onde o classificador apresentou um *F1-Score* abaixo de 60%.

4. Considerações Finais

Neste artigo, um novo sistema de apoio à decisão foi proposto para análise, detecção e diagnóstico de falhas em sistemas elétricos. O sistema combina técnicas de Aprendizado de Máquina e Lógica *Fuzzy* para classificar o estado atual da planta elétrica para depois disso gerar regras mais simples e interpretáveis para o especialista.

O sistema é composto por dois módulos: Módulo de Classificação e Módulo de Regras. O primeiro módulo emprega uma Árvore de Decisão com o objetivo de gerar a classificação do sistema juntamente com um conjunto de regras com valores *crisp* para qualquer conjunto de entradas. Após isso, o Módulo de Regras pode ser acessado pelo especialista do sistema para criação de variáveis *fuzzy* visando simplificar o conjunto de regras, aumentando sua interpretabilidade. Esse processo é interativo e pode ser ajustado em qualquer momento. O processo de construção de regras *fuzzy* é extremamente importante, sobretudo em contextos que podem existir inúmeras variáveis e regras complexas que podem ser importantes para diagnosticar e explicar os estados do sistema.

A avaliação experimental do Módulo de Classificação mostrou que sistema proposto empregando a Árvore de Decisão alcançou resultados superiores quando comparado com outros modelos da literatura (XGBoost, LightGBM e CatBoost) utilizando quatro métricas bem conhecidas na literatura. Os experimentos foram realizados com uma base de dados de falhas elétricas bem conhecida.

A adoção de um modelo *white-box* como classificador foi importante porque permitiu que as regras geradas fossem posteriormente tratadas pelo especialista com mais facilidade no Módulo de Regras. Assim, o conhecimento do sistema pode ser ampliado possibilitando regras mais simples, interpretáveis e inteligíveis fossem geradas, aumentando a consciência situacional da planta elétrica por parte do especialista.

Como trabalhos futuros, outros classificadores *white-box* [Martins et al. 2023, Zhu et al. 2022] devem ser avaliados em mais bases de dados de falhas de sistemas elétricos. Ainda visando aumentar a taxa de acerto do Módulo de Classificação, estratégias de *Ensembles* [Zhu et al. 2023] e seleção dinâmica de classificadores [Cruz et al. 2018] devem ser avaliada. Para o Módulo de Regras, formas de

simplificação de regras devem ser investigadas, além de analisar outras técnicas para gerar *insights* sobre a consciência situacional do sistema elétrico sob análise, como valor Shapley [Lee et al. 2023].

Agradecimentos

Os autores gostariam de agradecer ao Programa de Pesquisa Desenvolvimento e Inovação da Agência Nacional de Energia Elétrica (PD&I ANEEL) e a EVOLTZ.

Referências

- Al-Sarem, M., Saeed, F., Boulila, W., Emara, A. H., Al-Mohameed, M., and Errais, M. (2021). Feature selection and classification using catboost method for improving the performance of predicting parkinson's disease. In *Advances on Smart and Soft Computing: Proceedings of ICACIn 2020*, pages 189–199. Springer.
- Cruz, R. M., Sabourin, R., and Cavalcanti, G. D. (2018). Dynamic classifier selection: Recent advances and perspectives. *Information Fusion*, 41:195–216.
- Fahim, S. R., Sarker, Y., Sarker, S. K., Sheikh, M. R. I., and Das, S. K. (2020). Self attention convolutional neural network with time series imaging based feature extraction for transmission line fault detection and classification. *Electric Power Systems Research*, 187:106437.
- Gadzinski, G. and Castello, A. (2022). Combining white box models, black box machines and human interventions for interpretable decision strategies. *Judgment and Decision Making*, 17(3):598–627.
- Goni, M. F., Nahiduzzaman, M., Anower, M., Rahman, M., Islam, M., Ahsan, M., Haider, J., and Shahjalal, M. (2023). Fast and Accurate Fault Detection and Classification in Transmission Lines using Extreme Learning Machine. *e-Prime - Advances in Electrical Engineering, Electronics and Energy*, 3:100107.
- Jamil, M., Sharma, S. K., and Singh, R. (2015). Fault detection and classification in electrical power transmission system using artificial neural network. *SpringerPlus*, 4(1):1–13.
- Janarthanam, K., Kamalesh, P., Basil, T. V., and Kovilpillai, A. K. J. (2022). Electrical Faults-Detection and Classification using Machine Learning. In *2022 International Conference on Electronics and Renewable Systems (ICEARS)*, pages 1289–1295.
- Lee, Y.-G., Oh, J.-Y., Kim, D., and Kim, G. (2023). Shap value-based feature importance analysis for short-term load forecasting. *Journal of Electrical Engineering & Technology*, 18(1):579–588.
- Liew, X. Y., Hameed, N., and Clos, J. (2021). An investigation of XGBoost-based algorithm for breast cancer classification. *Machine Learning with Applications*, 6:100154.
- Martins, V. E., Cano, A., and Junior, S. B. (2023). Meta-learning for dynamic tuning of active learning on stream classification. *Pattern Recognition*, 138:109359.
- MathWorks (2022). Matlab version: 9.13.0 (r2022b).

- Mukherjee, A., Kundu, P. K., and Das, A. (2020). Application of principal component analysis for fault classification in transmission line with ratio-based method and probabilistic neural network: a comparative analysis. *Journal of The Institution of Engineers (India): Series B*, 101:321–333.
- Omar, A. M. S., Osman, M. K., Ibrahim, M. N., Hussain, Z., and Abidin, A. F. (2020). Fault classification on transmission line using LSTM network. *Indonesian Journal of Electrical Engineering and Computer Science*, 20(1):231–238.
- Pan, H., Li, Z., Tian, C., Wang, L., Fu, Y., Qin, X., and Liu, F. (2023). The lightgbm-based classification algorithm for chinese characters speech imagery bci system. *Cognitive Neurodynamics*, 17(2):373–384.
- Sagi, O. and Rokach, L. (2021). Approximating XGBoost with an interpretable decision tree. *Information Sciences*, 572:522–542.
- Samantaray, S., Dash, P., and Panda, G. (2007). Distance relaying for transmission line using support vector machine and radial basis function neural network. *International Journal of Electrical Power & Energy Systems*, 29(7):551–556.
- Toliat, H. A., Sadeh, J., and Ghazi, R. (1996). Design of augmented fuzzy logic power system stabilizers to enhance power systems stability. *IEEE Transactions on Energy conversion*, 11(1):97–103.
- Toraih, E. A., Elshazli, R. M., Hussein, M. H., Elgaml, A., Amin, M., El-Mowafy, M., El-Mesery, M., Ellythy, A., Duchesne, J., Killackey, M. T., et al. (2020). Association of cardiac biomarkers and comorbidities with increased mortality, severity, and cardiac injury in COVID-19 patients: a meta-regression and decision tree analysis. *Journal of medical virology*, 92(11):2473–2488.
- Uppendar, J., Gupta, C., and Singh, G. (2008). Discrete wavelet transform and genetic algorithm based fault classification of transmission systems. In *15th National Power Systems Conference, IIT Bombay*, pages 323–328.
- Wilfling, S., Ebrahimi, M., Alfalouji, Q., Schweiger, G., and Basirat, M. (2022). Learning non-linear white-box predictors: A use case in energy systems. In *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 507–512.
- Zhu, X., Li, J., Ren, J., Wang, J., and Wang, G. (2023). Dynamic ensemble learning for multi-label classification. *Information Sciences*, 623:94–111.
- Zhu, Y., Ma, J., Yuan, C., and Zhu, X. (2022). Interpretable learning based dynamic graph convolutional networks for alzheimer’s disease analysis. *Information Fusion*, 77:53–61.