IPSW 2023 Hitachi I Final Presentation

Efficient decision-making of relays through machine learning

Golshid Aflaki (HEC), Sean Bohun (OntarioTech), Faramarz Farhangian (ETS), Kirill Golubnichiy (University of Calgary), Nicolas Goulet (UQAM), Jean-François Plante (HEC), Maksym Shpakovych (Inria), Jhelum Chakravorty (Hitachi Energy)

August 25, 2023

Contents

Problem definition

Let us consider an electrical power line where accidents occur due to several factors, including:

- Internal fault
- Phase to ground fault
- Bushing explosion
- Circuit breaker failure
- Intermittent / Permanent earth fault
- Broken wire fault

Questions:

1. Can we learn, in an automatic fashion, the weights in the loss function to rebalance the dataset?
2. Can we decide when to trip the line due to the fault?
3. Can the location of the fault be determined?

Requirements

Power System Requirements:

- Speed – isolate fault as soon as possible to limit equipment damage
- Selectivity – isolate only faulted section of the power system
- Sensitivity / Security – isolate every fault
- Reliability / Dependability – only act when appropriate

'Classical' techniques:

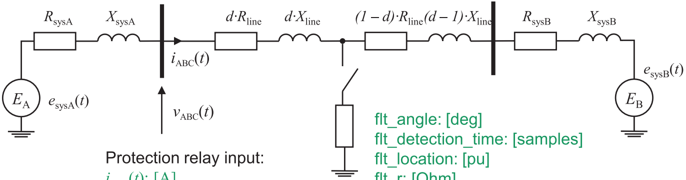
- Phasor measurements
- Unified impedance
- Fuzzy inference
- Wavelets
- Artificial neural networks

Data explanation

line_len: [km]
line_r0: [Ohm/km]
line_r1: [Ohm/km]
line_x0: [Ohm/km]
line_x1: [Ohm/km]
line_c0: [F/km]
line_c1: [F/km]

sys_b_r0 [Ohm]
sys_b_r1 [Ohm]
sys_b_sir: [pu]
sys_b_x0: [Ohm]
sys_b_x1: [Ohm]
sys_b_z0z1_ratio: [pu]
load_angle: [deg]

sys_a_V_nom: [V]
sys_a_r0 [Ohm]
sys_a_r1 [Ohm]
sys_a_sir: [pu]
sys_a_x0: [Ohm]
sys_a_x1: [Ohm]
sys_a_z0z1_ratio: [pu]



Protection relay input:
$i_{\text{ABC}}(t)$: [A]
$v_{\text{ABC}}(t)$: [V]
line_r0: [Ohm]
line_r1: [Ohm]
line_x0: [Ohm]
line_x1: [Ohm]
reach_setting: [pu] (default 0.7)

flt_angle: [deg]
flt_detection_time: [samples]
flt_location: [pu]
flt_r: [Ohm]

Machine learning (ML)

What is the task?
Make a sequential decision using only past samples at each step to decide if the circuit breaker should trip a fault within the protected zone.

Challenges:

- Speed of the decision making (fewer time samples)
- Control false-positive and false-negative rates (accuracy)

A potential ML based solution comprises the following steps:

- Fault detection within a few samples after occurrence of fault
- Reach setting (protected zone) information
- Binary decision of trip (1) or restrain (0)

Classification approach

- One way for solving the initial problem is to build a classification model $g_\theta : \mathbb{R}^{n+1} \mapsto [0, 1]$ returns the probability of a trip event.
- To make this model independent from the size of the protected zone, we propose to use it as an additional input, which gives $n + 1$ variables
- Use MLP for this problem which is trained to minimize the binary cross-entropy loss
- **Frequency-based** approach for rebalancing is applied. Other choices:
  - Li, M. et al. (2021). Autobalance: Optimized loss functions for imbalanced data. Advances in Neural Information Processing Systems, 34, 3163-3177.
  - Mukhoti, J. et al. (2020). Calibrating deep neural networks using focal loss. Advances in Neural Information Processing Systems, 33, 15288-15299.
  - Sinha, S. et al. (2020). Class-wise difficulty-balanced loss for solving class-imbalance. In Proceedings of the Asian conference on computer vision.

Classification approach

- The trained model has an F1 score of 0.98 on the validation set, indicating that the model has captured the dependencies between variables
- The probability returned by the model gives the possibility to control the false positive rate by using a different decision threshold

False positive rate	Threshold
1.06%	0.4
0.97%	0.5
0.89%	0.6

Regression approach

Another way of deciding whether to trip is to predict the location $y \in [0, 1]$ of the fault on the electrical line, and check that it is in the protected zone using information about the line such as a current, voltage, resistance, reactance, etc.

- This can be done with the regression model, which assumes the relationship between the line parameters $x \in \mathbb{R}^n$ and y of the form

$$y = f_\theta(x) + \varepsilon, \tag{1}$$

where $f_\theta : \mathbb{R}^n \mapsto \mathbb{R}$ is probably a nonlinear parametric function, θ is the parameters of f, and $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ is a random variable that captures the noise in the data for $\sigma > 0$

Regression approach

- In this workshop, we proposed to use a multi-layer perception with 3 layers of size 64 and ReLU activation as a function f in (1), trained to minimize the Mean Squared Error between its output $\hat{y}$ and y

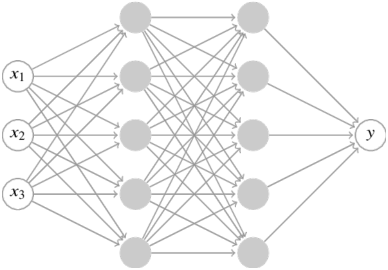


Figure: Diagram of the multilayer perceptron with 2 hidden layers

- We selected this type of model among others such as linear regression and decision trees since it gives the best score

Regression approach

- In the first experiments, we considered that among others x contains 10 values of current and voltage recorded after the fault was detected.
- The trained model was tested on the validation set and the following scores were obtained:

$$R^2 = 0.99, \qquad \frac{\|\hat{y} - y\|}{\|y\|} = 0.02,$$

  where R^2 is a coefficient of determination.
- Considering these values we can state that the MLP model captured the nonlinear dependence between the variables.

Regression approach

- The important requirement of the model is to have a low false positive rate, i.e. to avoid the case where we trip when it is not necessary.
- The trained model has 0.57% false positive rate on average.

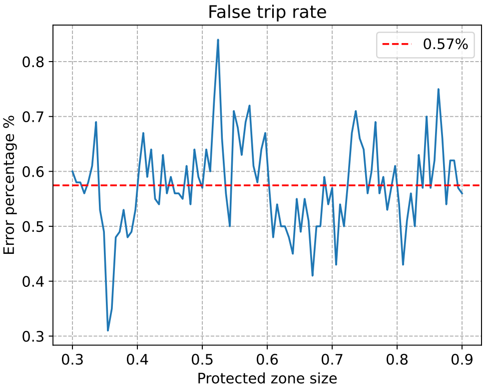


Figure: False positive rate as a function of protection zone size.

Utility function

To evaluate the regression model in this problem, it is not only the performance of the model that matters. but also we need to penalize the model for using more than a certain number of time samples (shift). To do so a penalty term is designed,

$$T = \begin{cases} 1, & \mathrm{t} \leq 5, \\ \dfrac{|t-30|^a}{25^a}, & 5 < t < 30, \\ 0, & t \geq 30, \end{cases} \tag{2}$$

in which t is the shift time samples after fault detection time, and a is a hyperparameter that adjusts the intensity of the penalty (default: $\mathrm{a} = 2$). The utility function then could be,

$$Utility = \frac{1}{RMSE} + T \qquad \text{or} \qquad Utility = \frac{T}{RMSE}.$$

We will continue with the utility based on summation.

Penalty term

The penalty term defined in 2 can be modified in different ways to penalize more or less based on certain thresholds,

$$T = \begin{cases} 1, & t \leq 5 \\ \dfrac{|t-30|^a}{25^a}, & 5 < t < 30 \\ 0, & t \geq 30 \end{cases} \qquad T = \begin{cases} 1, & t \leq 5 \\ \dfrac{|t-20|^a}{15^a}, & 5 < t < 20 \\ 0, & t \geq 20 \end{cases} \qquad T = \begin{cases} 1, & t \leq 5 \\ \dfrac{|t-10|^a}{5^a}, & 5 < t < 10 \\ 0, & t \geq 10 \end{cases}$$

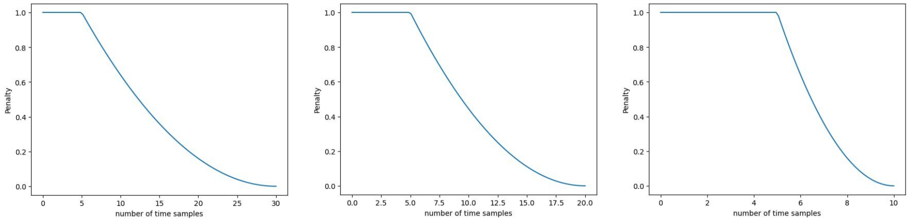
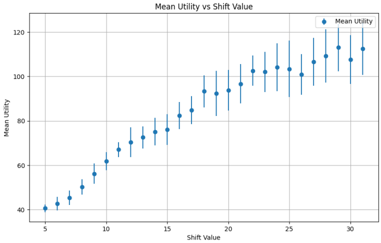


Figure: In the left plot the penalty term is forced to be zero by 30 number of time samples, in the middle plot, the penalty term is forced to zero by 20 time samples, and the last plot shows at 10 time samples the penalty term is forced to zero

Model evaluation

We want to evaluate the performance of our regression model,

- 27 models are trained based on shifts between 5 to 31.
- 20 replications is considered, the average RMSE and the average utility is computed.



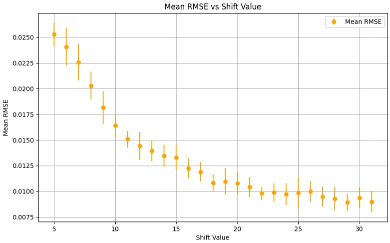


Figure: The left plot shows the average utility for different shift values, and the right plot shows the average RMSE for different shift values

Mean and variance trade-off for different shifts

Based on the results of the 27 models with 20 replications, we want to pick the optimal number of time samples (shifts) in a way that increases the performance of the model while we have as less as possible shifts making sure that we have enough information. In the following plot, we are looking for,

- Higher Utility with low variance
- Lower RMSE with low variance

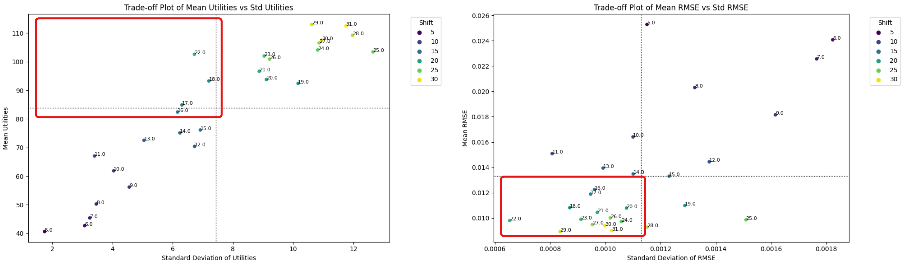


Figure: trade-off between mean and variances in Utility (left) and RMSE (right)

Conclusion and future work

In conclusion, we have two approaches to address the challenges of accuracy and timeliness of the binary decision of the relay, one is the classification, and the other one is the regression approach. In the classification task, we are faced with the class-imbalance problem which is not the case in the regression task.

Our analysis during this workshop indeed has its limitations, to improve the basic ideas presented we can try the following:

- have more than 20 replications for more robust results.
- consider different classification loss functions to dynamically rebalance classes, such as focal loss, and dynamically learning the weights as a function of reach setting and fault distribution, etc.
- consider a post-processing analysis such as calibration and cost-sensitive learning of a sufficient number of time samples.
- consider dynamic modeling decision-making.
- explore the evolution of a Kalman filter as a fault is initiated.