

# Genetic programming for shunt failure detection in monophasic transmission line

Edgar A. Gamboa Rodriguez\* Cristina Verde Rodarte.\*

\* *Instituto de Ingeniería, Universidad Nacional Autónoma de México, México. (e-mail: edgararturogamboa@comunidad.unam.mx, Cverde@unam.mx).*

---

**Abstract:** In this work, the usage of bio-inspired algorithms for developing models capable of identifying the ubication of the anomaly, specifically a shunt fault in High Voltage Alternate Current (HVAC) for a monophasic transmission line. The approach is data-driven based, where the line is simulated under different operation conditions, afterward, the cross-correlation is used for extracting the most representative characteristics of the signals. The specifications under which the genetic algorithm searched for a solution, only with the correlations, generating mathematical models are able of identifying the distance of the fault even for changing conditions. Finally, the model found by the genetic program is validated through two different sets of data (training and test), having different combinations of parameters and measuring the average error between the predicted and the real distance.

**Keywords:** Genetic Programming, AC transmission line, Shunt Failure, Failure diagnosis, Signal cross-correlation.

---

## 1. INTRODUCCIÓN

Uno de los grandes problemas que afectan el funcionamiento de las líneas de transmisión eléctrica, tanto de corriente alterna (AC) como de corriente directa (DC), son las fallas de derivación (Shunt fault). Estas involucran una combinación diferente entre las 3 fases y la tierra. Pero la falla más común entre estas es la de una línea a tierra (LG), representando hasta un 85 % del total de este tipo de fallas (Kunj et al., 2019).

En la literatura se ha explorado el uso de metodologías basados en datos (Data-driven) con lo cual es posible aplicar diferentes modelos de inteligencia artificial (AI) para la identificación de fallas, en Yu and Zhang (2023) se hace uso de técnicas de AI como máquinas de soporte de vectores (SVM) hasta el uso de redes neuronales artificiales (ANN).

Es posible plantear el problema de forma categórica, en conjunto técnicas de AI y extracción de características se puede dar una solución al problema con un alcance más limitado. Como se observa en el trabajo de Manohar et al. (2016) donde mediante el uso de la transformada para extracción de características y el uso de SVM o de ANN se obtiene un 100 % de precisión para ambos casos, y al considerar el uso de ruido el desempeño baja hasta un 92 % y 93 % respectivamente. En Tekli et al. (2013) usando el algoritmo de entrenamiento de Levenberg-Marquardt para una red neuronal obtuvieron un error promedio de 6.6 % para la ubicación de la falla y 4.3 %

para el valor de la resistencia. En Jamil et al. (2014) se compara el desempeño de una red neuronal generalizada (GNN) con un error medio del 2 % contra una ANN que obtuvo un desempeño del 3 % de error respecto al tamaño de la línea.

Los algoritmos genéticos (GA) son algoritmos que tratan de replicar el comportamiento de evolución de los seres vivos para la toma de decisiones de optimización e identificación de modelos. Destacan en problemas de optimización, así como su aplicación en una gran variedad de campos del conocimiento (Valdez et al., 2021). En el área de detección de fallas donde se identifica la presencia de eventos anormales y deterioro de los componentes de los procesos de forma exacta y con cierta rapidez. El uso de estas técnicas bio-inspiradas destacan cuando se buscan aquellos parámetros, que otorgan el mejor desempeño con las diferentes técnicas de AI (Zhang et al., 2005).

Programación genética (GP) se puede considerar una extensión de los algoritmos genéticos, estos difieren en que no tienen un codificación lineal si no forma de árbol. Esta nueva representación planteado por Koza (1994) otorga mayor flexibilidad al momento de explorar el espacio de búsqueda de soluciones. En particular para diagnosticar fallas en líneas de transmisión se plantea el uso de funciones matemáticas generadas a partir de la GP con la finalidad de identificar la ubicación de la ocurrencia de la falla de derivación.

La presentación de este trabajo esta organizado de la siguiente forma, la Sección 2 introduce el procedimiento

usado acerca de cómo y de qué forma fueron generados los datos y que circuito eléctrico se considera; la Sección 3 presenta de forma resumida los pasos necesarios para la extracción de características de las señales generadas. La Sección 4 habla acerca de los parámetros más relevantes, nodos funciones y nodos terminales usados en el algoritmo genético que produjeron los resultados; la Sección 5 expone la mejor función encontrada, así como el desempeño en el conjunto de entrenamiento y validación. Finalmente la Sección 6 indaga en las conclusiones del trabajo así como posibles mejoras para un trabajo futuro.

## 2. MODELADO DE LÍNEA Y GENERACIÓN DE DATOS

En esta sección se presenta el modelo usado para la generación de los datos de una HVAC monofásica. El modelo matemático de la línea en condición normal es el par de ecuaciones diferenciales parciales conocidas como las ecuaciones telegrafista. Sin embargo dado que el procedimiento que se presenta no hace uso de ellas, se omite por una de espacio  $E(s)$  Esta línea trasmite la corriente de desde una fuente ( $E_t$ ) hasta una carga resistiva ( $R_l$ ) a través de una distancia ( $l$ ). La línea está constituida por una secuencia continua e infinitesimal de elementos filtrantes compuestos por resistencia, inductancia, capacitancia y conductancia, denotados como RLCG. (Gönen, 2015).

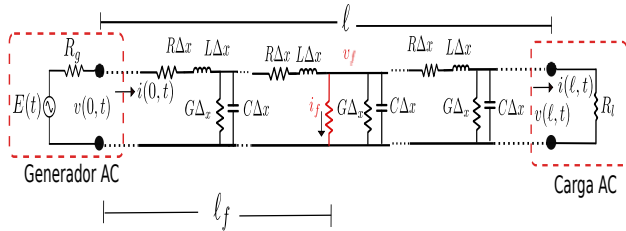


Figura 1. Modelo de línea de transmisión con una falla en derivación a una distancia  $l_f$  del generador

A partir del modelo en la Figura (1) se produce una falla en derivación que ocurre a una distancia  $l_f$ .

A través de la utilización de ATPDRAW, un software de simulación de sistemas eléctricos de potencia, se generó una base de datos de entrenamiento. Esta base de datos simula una falla de derivación de línea a tierra mediante la asignación de valores constantes detallados en la tabla presentada 1.

En cada iteración de la simulación se elige una combinación entre distancia  $l_f$ , resistencia  $R_f$  y el desfase de señal.

El conjunto de datos de prueba se uso la misma combinación de parámetros fijos de la Tabla 1 y variables de resistencia  $R_f$  de la Tabla 3 y desfase en la señal de la Tabla 4, cambiando el conjunto de distancias de entrenamiento de la Tabla 2 por el de validación de la Tabla 5.

Parámetro	Valor
Resistencia $R$	$1.33 \times 10^{-6} [\Omega/m]$
Capacitancia $C$	$8.86 \times 10^{-12} [F/m]$
Inductancia $L$	$9.3 \times 10^{-4} [H/m]$
Admitancia $G$	$NA$
Frecuencia	60 [Hz]
Longitud de la línea $l$	300 [km]
Voltaje Nominal $E(t)$	220 [kV]
Muestreo	10 [kHz]

Tabla 1. Parámetros de la línea (Indulkar and Ramalingam, 2008)

Distancia $l_f$				
10 [km]	20 [km]	...	280 [km]	290 [km]

Tabla 2. Distancia de entrenamiento

Resistencia $R_f$				
10 [ $\Omega/m$ ]	20 [ $\Omega/m$ ]	...	90 [ $\Omega/m$ ]	100 [ $\Omega/m$ ]

Tabla 3.  $r$  de falla

Desfase de la señal			
$0^\circ$	$90^\circ$	$180^\circ$	$270^\circ$

Tabla 4. Desfase de la señal en grados

Distancia $l_f$				
5 [km]	15 [km]	...	285 [km]	295 [km]

Tabla 5. Distancia de prueba

## 3. PROCESAMIENTO DE DATOS

En esta sección se resume cual fue el proceso utilizado para la extracción de características asociadas al modelo simulado de la línea HVAC con la falla a una distancia  $l_f$ .

Partiendo de las señales generadas, en los observadores en los extremos de la línea se mide la corriente  $i(0, t)$  e  $i(l, t)$ , se toma una ventana de interés de 50 [ms], donde a los 25 [ms] ocurre la falla. Se realiza una división en tres partes de la señal de entrada y salida separadas a intervalos de 1 ciclo (1 [Hz]) de la señal como se observa en la Figura 2. Cada ciclo representa:

1. Estado estable de las señales antes que ocurra la falla.
2. Ciclo donde se presenta la falla, conteniendo los cambios durante el estado transitorio.
3. Estado estable de ambas señales una vez ya ocurrió la falla.

Usando las señales segmentadas se utiliza la correlación cruzada sin normalizar Eq. (1) entre la señal de entrada y la de salida para cada uno de los bloques (Stoica and Moses, 2005).

$$\hat{r}_{xy}(m) = \begin{cases} \sum_{t=k+1}^N y(t)u^*(t-k), & k \geq 0, \\ \hat{r}_{xy}^*(-m), & k < 0, \end{cases} \quad (1)$$

Generado las tres correlaciones siguientes:

Observando la figura 3 las correlaciones cruzadas antes (3a) y después (3b) de la falla tienen una forma muy similar entre si, mientras que la que correlación durante el

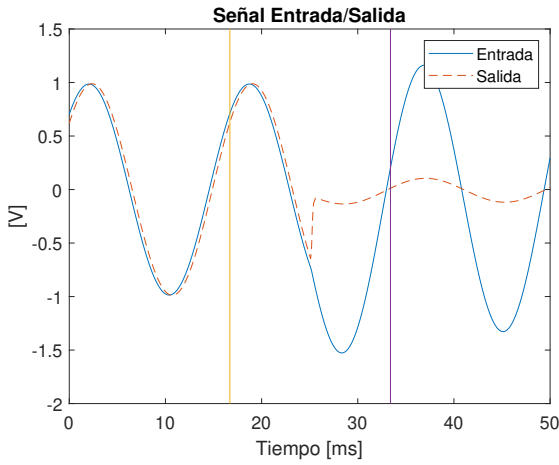


Figura 2. Señales de entrada y salida

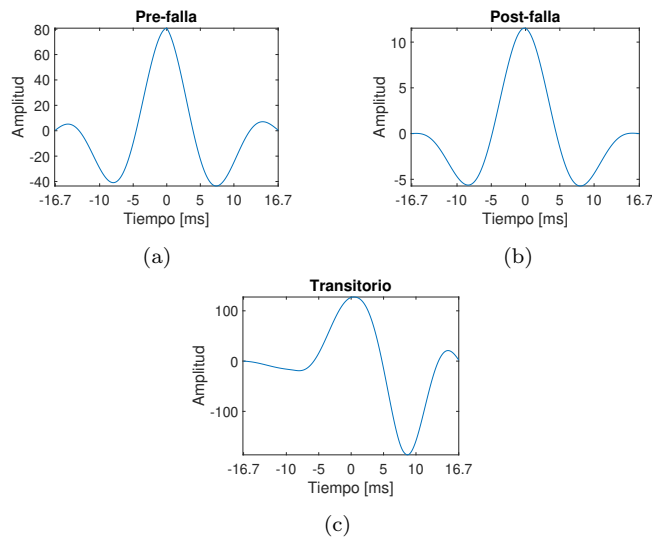


Figura 3. Correlación cruzada de los diferentes ciclos de las señales

estado transitorio (3c) tiene un patrón diferente. Extrapolando esta relación de correlación a lo largo de diversas distancias, se puede notar que la amplitud exhibe una tendencia inversamente proporcional a la distancia, tal como se ilustra en la Figura 4.

Dado que se conoce el funcionamiento normal de operación de la línea de transmisión, a partir de los datos de entrada se puede predecir el comportamiento del sistema Ec.(2) es similar al comportamiento observado mediante otros observadores

$$\hat{Y}(s) = G(s)U(s) \quad (2)$$

Al momento de que se presenta una falla el modelo de predicción sufre un cambio Ec. (3) lo que produce predicciones diferentes a las que están siendo observadas (Verde et al. (2012))

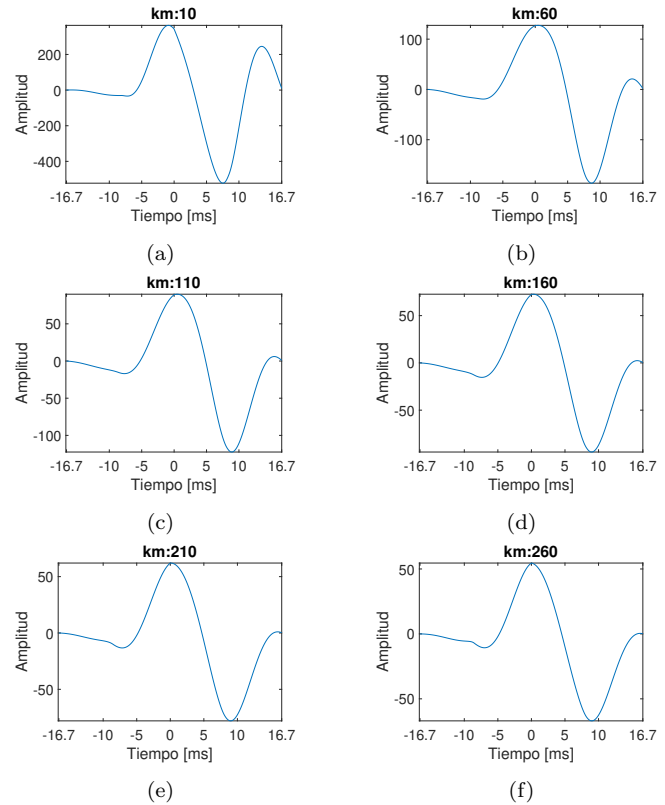


Figura 4. Correlación cruzada del estado transitorio en diferentes distancias  $l_f$

$$\hat{Y}(s) = (G(s) + \Delta_s)U(s) \quad (3)$$

El método de caracterización propuesto para la predicción de la distancia  $l_f$  utilizará la correlación del estado transitorio (Figura 3c), la cual extrae la información del cambio producido por la falla de derivación. En este enfoque, se identificarán los dos valores más elevados y los dos valores más bajos para su posterior análisis de la señal, y cada uno tiene relacionado la distancia  $l_f$  donde ocurrió la falla.

#### 4. PROGRAMACIÓN GENÉTICA

En esta sección se describe los parámetros usados con la programación genética así como una breve descripción de en que consiste dicha configuración. Los individuos se representan mediante árboles binarios, donde los nodos función y los nodos terminales definen las operaciones y operandos necesarios para resolver el problema en cuestión.

Cada individuo representa una posible solución al problema, el cual es evaluado para obtener su desempeño y ser comparado con el resto de la población. La Figura 5 representa un individuo el cual es transformado en la Ec. (4) para ser evaluado.

$$f(x_1, x_2, x_3, x_4) = x_1 - 20x_4 + \frac{SIN(x_3)}{x_2} \quad (4)$$

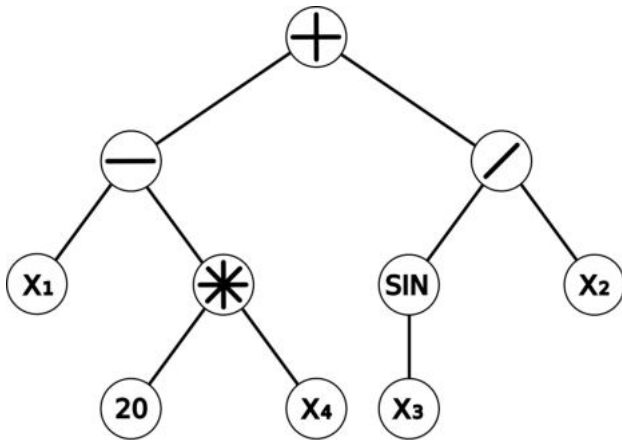


Figura 5. Árbol de programación genética

Para que el algoritmo fuera capaz de encontrar una solución que tuviera un buen desempeño en el problema de detección, los parámetros que se usaron fueron:

- Población: Se usan 50 individuos inicializados de forma aleatoria.
- Tamaño individuo: El tamaño máximo del árbol binario es hasta 1024 nodos.
- Evaluación: La función de aptitud (fitness) compara el resultado predicho contra el real, sumando todas las evaluaciones para obtener el desempeño de un individuo.

$$fitness = \sum_{i=0}^n (y_i - \hat{y}_i)^2 \quad (5)$$

- Selección: Una vez se han evaluado los individuos estos se enfrentan mediante torneo contra otro de forma aleatoria y se selecciona como padre a aquel que haya obtenido mejor desempeño.
- Cruza: Se refiere a la probabilidad con la que dos padres intercambian subárboles entre si, esta se realiza en un solo punto con probabilidad del 95 %.
- Mutación: Probabilidad del 1 % que un subárbol de un individuo sea cambiado por uno diferente aleatoriamente.
- Elitismo: El mejor individuo de cada generación reemplazara al individuo con peor desempeño de la próxima generación, en caso de empate será seleccionado el individuo de menor tamaño.
- Terminales: El conjunto de terminales son aquellos que forman los nodos hojas del árbol, los valores usados fueron:

- $x_1$
- $x_2$
- $x_3$
- $x_4$
- $-1$
- $1$
- $5$

- 10
- 20
- 50
- 100

Donde cada  $x$  representa un punto característico extraído de la correlación del estado transitorio (Figura 3c).

El conjunto de nodos terminales fue seleccionado a partir de experimentación donde valores más pequeños no permitían que el algoritmo obtuviera soluciones rápido, y valores más grandes terminaban por no estar presentes.

- Funciones: Son los nodos que marcan que operaciones se deben de realizar entre los nodos hijos del árbol. Las funciones usadas son:

- Suma:

$$SUM(x, y) = x + y \quad (6)$$

- Resta:

$$LESS(x, y) = x - y \quad (7)$$

- Multiplicación:

$$MUL(x, y) = x * y \quad (8)$$

- División:

$$DIV(x, y) = \begin{cases} \frac{x}{y}, & y \neq 0, \\ \frac{x}{0.0001}, & y = 0, \end{cases} \quad (9)$$

- Seno:

$$SIN(x) = \sin x \quad (10)$$

- Absoluto:

$$ABS(x) = |x| \quad (11)$$

- Polinomio:

$$POLY(x, y) = x^2 + 2xy + y^2 \quad (12)$$

El algoritmo genético inicialmente empleó cinco funciones fundamentales la suma, resta, multiplicación, división protegida y la función seno. Mediante un análisis de las soluciones obtenidas, se identificó la existencia de bloques específicos, los cuales exhiben una estructura clara, bien definida y se encuentran en diversas ubicaciones dentro del árbol. Posteriormente, se incorporaron dos funciones adicionales al conjunto la función de valor absoluto y la función cuadrática.

## 5. RESULTADOS

En este apartado se muestra el mejor resultado encontrado tras varias ejecuciones de algoritmo bajo las mismas condiciones de operación. El mejor resultado encontrado usando el conjunto de entrenamiento por el algoritmo fue:

$$SUM(POLY(ABS(ABS(DIV(SIN(20), ABS(x_3))))), ABS(MUL(DIV(1, DIV(LESS(x_2, -1), 20)), 20))), POLY(ABS(MUL(DIV(1, DIV(LESS(DIV(MUL(MUL(ABS(x_3), 5), x_3), MUL(POLY(DIV(50, x_1), SUM(50, x_1)), LESS(100, LESS(1, x_4))))), -1), 10)), DIV(1, 1))), ABS(MUL(DIV(1, DIV(LESS(x_2, -1), 20)), 20))))).$$

A partir del cual se puede simplificar y reducir términos redundantes quedando la siguiente expresión:

$$x_{eq} = \frac{x_3^2}{2 \left( \frac{50}{x_1} + x_1 + 50 \right)^2 (100 + x_4)} \quad (13)$$

$$f(x_1, x_2, x_3, x_4) = \left( \left( \frac{1}{x_3} + \frac{400}{x_2} \right)^2 + \left( \left( x_{eq} + \frac{1}{10} \right)^{-1} + \left| \frac{400}{x_2} \right| \right)^2 \right)^2 \quad (14)$$

En relación al conjunto de datos de entrenamiento, el modelo exhibe una tendencia de aproximación como se ilustra en la Figura 6, oscilando en torno al valor real de la ubicación de la falla de derivación, con un error promedio del 3.15 %.

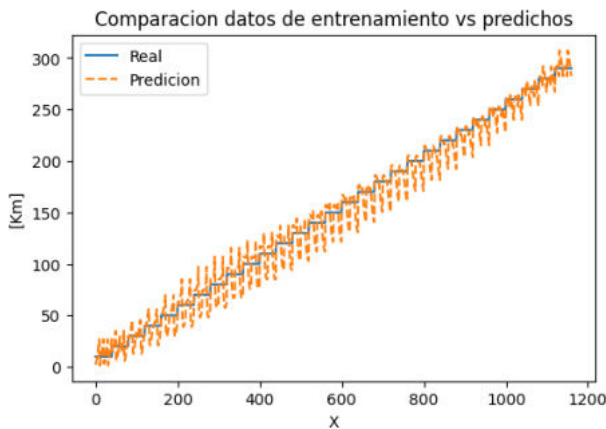


Figura 6. Predicción con datos de entrenamiento

En cambio se observa en el conjunto de prueba mostrado en la Figura 7 un mejor desempeño en el conjunto de datos de prueba, obteniendo un error promedio de 7.87 [Km] respecto a la ubicación de la falla, es decir, un error del 2.62 % error respecto al tamaño de la línea.

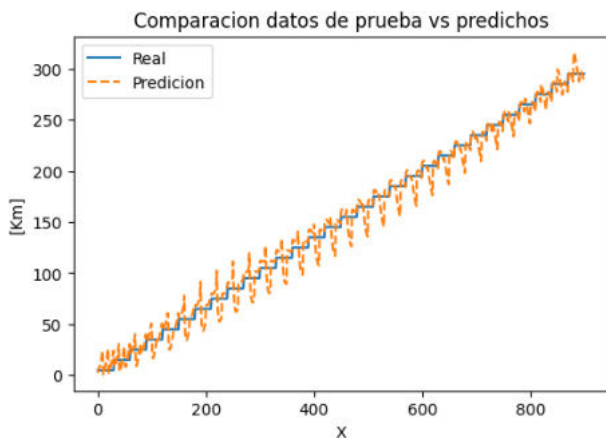


Figura 7. Predicción con datos de prueba

## 6. CONCLUSIONES

En este trabajo se abordó el uso de algoritmos bio inspirados como generadores de funciones para el diagnóstico de fallas de derivación en líneas de transmisión eléctrica de corriente alterna monofásicas. Con la finalidad de producir un modelo matemático, se usó un enfoque basado en datos. Gracias al uso del modelo simulado mediante el software ATPDRAW, se obtuvieron las observaciones de las señales de la corriente tanto en la entrada como a la salida de la línea.

A partir de estas simulaciones se procedió a extraer las características más representativas que fueron usadas durante la programación genética. El uso de la correlación cruzada entre la señal de entrada y salida demostró ser una señal lo suficientemente rica en datos para que el algoritmo de optimización pudiera encontrar soluciones que se puedan aproximar a la distancia objetivo.

La mejor solución encontrada tiene un buen desempeño en el conjunto de entrenamiento y mejorando levemente en el conjunto de prueba. Al producir una solución compuesta de funciones sencillas el tiempo necesario para la ejecución de el algoritmo es rápida, con un error de estimación del 3 %. A pesar de tener una gran variabilidad en el error de la respuesta dada, la metodología propuesta mostró gran potencial para dar soluciones a este tipo de problemas.

Al tratarse de algoritmos basados en la evolución, múltiples mejoras se pueden implementar para reducir el error entre la distancia real y la generada:

- Dado que la generación, cruza y mutación de los individuos es aleatoria, ejecutar más veces el algoritmo permite encontrar diferentes resultados.
- Se usaron 500 generaciones para determinar cual había sido la mejor solución, por lo que es viable aumentar el tiempo de búsqueda incrementando las generaciones.
- Las funciones ayudan a que el algoritmo encuentre soluciones más aptas, por lo que agregar más opciones que se ajusten más a este tipo de problema, mejorarán la calidad de las respuestas encontradas, y de igual forma reducen el tamaño de los individuos.

## REFERENCIAS

- Gönen, T. (2015). *Electrical Power Transmission System Engineering: Analysis and Design, Third Edition*. doi: 10.1201/b17055.
- Indulkar, C.S. and Ramalingam, K. (2008). Estimation of transmission line parameters from measurements. *International Journal of Electrical Power and Energy Systems*, 30. doi:10.1016/j.ijepes.2007.08.003.
- Jamil, M., Kalam, A., Ansari, A., and Rizwan, M. (2014). Generalized neural network and wavelet transform based approach for fault location estimation of a transmission line. *Applied Soft Computing*, 19, 322–332. doi:https://doi.org/10.1016/j.asoc.2014.02.020. URL

- <https://www.sciencedirect.com/science/article/pii/S1568494614000908>.
- Koza, J.R. (1994). Genetic programming as a means for programming computers by natural selection. *Statistics and Computing*, 4. doi:10.1007/BF00175355.
- Kunj, T., Ansari, M.A., and Vishwakarma, C.B. (2019). Transmission line fault detection and classification by using wavelet multiresolution analysis: A review. *2018 International Conference on Power Energy, Environment and Intelligent Control, PEEIC 2018*. doi: 10.1109/PEEIC.2018.8665477.
- Manohar, M., Pallavi, R., and kumar, R.P. (2016). A universal high impedance fault detection technique for distribution system using s-transform and pattern recognition. *Technology and Economics of Smart Grids and Sustainable Energy*, 1, 9. doi:10.1007/s40866-016-0011-4. URL <https://doi.org/10.1007/s40866-016-0011-4>.
- Stoica, P. and Moses, R.L. (2005). *Spectral analysis of signals*. Pearson/Prentice Hall.
- Tekli, L., Filipovi-Gri, B., and Pavii, I. (2013). Artificial neural network approach for locating faults in power transmission system. *EuroCon*. doi: 10.1109/eurocon.2013.6625165.
- Valdez, F., Castillo, O., and Melin, P. (2021). Bio-inspired algorithms and its applications for optimization in fuzzy clustering. *Algorithms*, 14. doi: 10.3390/a14040122.
- Verde, C., Gentil, S., and Morales-Menendez, R. (2012). *Monitoreo y Diagnóstico Automático de Fallas en Sistemas Dinámicos*.
- Yu, J. and Zhang, Y. (2023). Challenges and opportunities of deep learning-based process fault detection and diagnosis: a review. *Neural Computing and Applications*, 35. doi:10.1007/s00521-022-08017-3.
- Zhang, L., Jack, L.B., and Nandi, A.K. (2005). Fault detection using genetic programming. *Mechanical Systems and Signal Processing*, 19. doi: 10.1016/j.ymsp.2004.03.002.