Universidad Tecnológica Nacional
DICIEMBRE 2020 / Año 18- Nº 39

Revista Tecnología y Ciencia
DOI: https://doi.org/10.33414/rtyc.39.85-102.2020 - ISSN 1666-6933

Regresión Simbólica aplicada a la Predicción del Consumo Eléctrico a Corto Plazo en el Nivel de Subestación

Symbolic Regression applied to the Short-Term Load Forecasting on the Substation level

Presentación: 08/07/2020

Aprobación: 28/09/2020

Victor A. Jimenez
Grupo de Investigación en Tecnologías Informáticas Avanzadas, Facultad Regional Tucumán, Universidad Tecnológica Nacional - Argentina
adrian.jimenez@gitia.org

Gonzalo E. Lescano
Centro de Investigación de Atmósfera Superior y Radiopropagación, Facultad Regional Tucumán, Universidad Tecnológica Nacional - Argentina
onzalo@gmail.com

Adrian Will
Grupo de Investigación en Tecnologías Informáticas Avanzadas, Facultad Regional Tucumán, Universidad Tecnológica Nacional - Argentina
adrian.will@gitia.org

Resumen

El modelado de datos es un problema fundamental en diversas áreas del conocimiento. La Regresión Simbólica es una técnica que permite encontrar una relación matemática para describir un conjunto de datos experimentales. A diferencia de los métodos tradicionales de modelado, la Programación Genética permite encontrar una expresión matemática susceptible de ser analizada e interpretada. Multi-Expression Programming es una variante de Programación Genética, que presenta múltiples ventajas, haciéndola apta para su uso en casos reales. En este trabajo proponemos aplicar dicha variante para descubrir modelos de predicción, un día hacia adelante, de la corriente eléctrica de fase de una subestación transformadora en Tucumán, Argentina. Para analizar el comportamiento del algoritmo y ajustar parámetros, se realizaron pruebas utilizando Benchmarks conocidos. Se concluye que Multi-Expression Programming es adecuada para encontrar modelos en problemas complejos, y en el caso de predicción de corriente eléctrica se logró un nivel de error similar al obtenido con otras técnicas.

Revista Tecnología y Ciencia
DOI:https://doi.org/10.33414/rtyc.39.85-102.2020 - ISSN 1666-6933

Palabras clave: Programación Genética Lineal, Regresión Simbólica, Predicción del consumo eléctrico, Subestaciones Transformadoras.

Abstract:

Data modeling is an important problem in several areas of knowledge. Symbolic Regression is a technique that allows finding a mathematical relationship to describe a set of experimental data. Unlike traditional modeling methods, Genetic Programming allows us to find a mathematical expression that can be analyzed and interpreted. Multi-Expression Programming is a variant of Linear Genetic Programming, which has many advantages, making it suitable for real cases. In this work, we propose to apply this variant of Genetic Programming to discover models for phase current forecasting, one day ahead, using data from a transformer substation located in the province of Tucumán, Argentina. First, multiple tests were performed using known Benchmark functions to analyze the algorithm's behavior and adjust parameters. We concluded that Multi-Expression Programming is adequate to find models in complex problems as short-term load forecasting, achieving a similar error level compared to other techniques.

Keywords: Linear Genetic Programming, Symbolic Regression, Short-Term Load Forecasting, Transformer Substations

INTRODUCCIÓN

El sistema eléctrico ha evolucionado a paso firme desde su creación, pero no a la velocidad con la que se fue dando el exponencial crecimiento de la demanda energética. Esto hace que el sistema actual resulte ineficiente en numerosas ocasiones (Sandineni & Boehm, 2012). Para mejorar esta situación, surge la necesidad de redefinir la nueva generación de redes de transmisión eléctrica, denominadas Smart-Grids (Troiano et al., 2016). Estas redes son auto-regenerativas, resistentes a anomalías, eficientes, confiables, y permiten obtener información sobre todas las etapas del uso de la energía. A través de la implementación de este tipo de redes eléctricas inteligentes es posible tener un mejor control sobre toda la red, permitiendo tomar mejores decisiones y actuar con mayor eficiencia, desperdiciando menos recursos, y minimizando costos y pérdidas.

El advenimiento de las Smart Grids (SGs) permite la recolección a tiempo real de datos sobre el estado de funcionamiento de la red eléctrica. En base a esta disponibilidad de datos, resulta factible la implementación de muchas técnicas de análisis de datos para optimizar la gestión de la distribución eléctrica y operaciones relacionadas (Government, 2011). Entre estas técnicas, la predicción del consumo eléctrico es uno de los enfoques más utilizados para respaldar la toma de decisiones relacionadas a tareas de planificación de asignación y mantenimiento. Particularmente, la predicción del consumo eléctrico a corto plazo, conocida como Short-Term Load Forecasting o STLF, juega un importante rol para las empresas del sector eléctrico, especialmente en lo que se refiere a distribución, debido a que permite tener un mejor panorama para tomar decisiones estratégicas y operativas (Palacio, 2001). Sin embargo, la predicción del consumo es un problema de difícil solución por distintos motivos. La demanda de electricidad es un proceso complejo no estacionario con fuertes componentes aleatorios, compuesto por la contribución de muchos nodos individuales (Sevlian & Rajagopal, 2014).

Revista Tecnología y Ciencia
DOI:https://doi.org/10.33414/rtyc.39.85-102.2020 ISSN 1666-6933

En cuanto a la aplicación de un sistema de predicción de consumo a nivel Subestación Transformadora (SET), resulta crítica debido a que el excesivo consumo produce serios daños a los equipos de transmisión y distribución de energía (Jimenez et al., 2017). Además, estos daños no sólo afectan a las empresas sino a los consumidores finales. Se produce intermitencia del servicio provocando malestar a los usuarios, y perjudicando la imagen de la empresa. Estos daños pueden ser eficientemente evitados si se conoce con antelación los periodos en los cuales se observa un fuerte aumento de la demanda, llamados Picos de Consumo (Salimi-beni et al., 2006). Es importante destacar que la reparación o reemplazo de estos equipos es un proceso lento y prohibitivamente costoso en las condiciones actuales de la provincia y el país, por lo que el desarrollo e implementación de este tipo de sistemas resulta de importancia estratégica en las actuales condiciones. En la literatura, el problema es conocido como predicción del consumo eléctrico, pero el mismo abarca la predicción de diferentes variables eléctrica como energía consumida, corriente eléctrica , consumo eléctrico, etc. En este trabajo nos enfocamos en la predicción de la corriente eléctrica en cada fase de una SET particular.

Tradicionalmente, los métodos utilizados para el STLF pueden dividirse en dos grupos. El primero corresponde a los métodos estadísticos, diseñados para obtener una función de regresión que estima los valores utilizando datos de carga histórica. Algunos de estos métodos son la Regresión Multilineal (Amral et al., 2007; Rothe et al., 2009; Tuaimah & Abass, 2014), Autoregressive Integrated Moving Average o ARIMA (Lee & Ko, 2011; Nataraja et al., 2012) y Exponential Smoothing (Abd Jalill et al., 2013). Los métodos del segundo grupo se basan en herramientas de Inteligencia Artificial, y se utilizan para superar algunas de las limitaciones que tienen los métodos estadísticos. Support Vector Regression o SVR (Ceperic et al., 2013) y las Redes Neuronales Artificiales (Borges et al., 2013) son los métodos más populares, principalmente porque pueden lograr una buena exactitud en las predicciones sin mucha experiencia. Una red neuronal de tipo Feedforward entrenadas con método de Backpropagation o BPNN (Ahmad, 2012; Kamboj & Avtar, 2013) es una de las arquitecturas más utilizadas por su capacidad de extraer relaciones complejas y no lineales de los datos. Por otro lado, las Redes Neuronales de Función de Base Radial o RBFNN tienen una arquitectura especial y pueden ser utilizadas para la predicción (Arora et al., 2014; Jimenez et al., 2017). Todos estos métodos tienen ciertos inconvenientes, como la difícil parametrización y la posibilidad de que se produzca un sobreajuste u Overfitting (Borges et al., 2013). Además, las redes BPNN utilizan técnicas de gradiente para el entrenamiento, lo que la hace más lenta y puede quedar atrapada en el óptimo local. Las RBFNN, en cambio, tienen una mejor capacidad de aproximación, estructuras más simples y un algoritmo de aprendizaje más rápido, pero la optimización de su estructura sigue siendo un trabajo difícil (Lee & Ko, 2009).

Las técnicas de modelado tradicionales mencionadas anteriormente funcionan como una "caja negra", donde realmente no queda explicitado el modelo. Esto provoca que los expertos en el área particular del problema afrontado sean reticentes a utilizarlos, ya que no logran comprenderlos en profundidad. La Regresión Simbólica, en cambio, es una técnica de modelado que pretende encontrar una relación matemática, específicamente una ecuación, que describa a los datos, tal que la misma permita explicarlos de la mejor forma posible. Es decir que, a partir de un conjunto de datos experimentales, se obtiene una expresión matemática que identifica su comportamiento susceptible de ser analizada e interpretada. La *Regresión Simbólica* es un tipo de problema abordado comúnmente con distintas variantes de Programación Genética (GP por si siglas en inglés), la cual es una metodología basada en algoritmos inspirados en la evolución biológica. La misma es una especialización de los

Revista Tecnología y Ciencia
DOI:https://doi.org/10.33414/rtyc.39.85-102.2020 - ISSN 1666-6933

algoritmos genéticos, que intenta encontrar soluciones a problemas a partir de la inducción de programas de forma automática, sin requerir que el usuario conozca o especifique la forma o estructura de la solución a encontrar (Koza, 1994).

Existe una gran variedad de problemas de regresión simbólica que pueden ser atacados con GP, entre ellos descubrir identidades trigonométricas (Jimenez et al., 2015; Oplatkova et al., 2018), modelado y predicción financiera (Aguilar-Rivera et al., 2015), descubrimiento empírico de leyes científicas (Elli et al., 2014), encontrar soluciones de ecuaciones diferenciales (La Cava et al., 2014), entre otros (Brameier & Banzhaf, 2007; Gandomi et al., 2015; Koza, 1994). Una revisión completa de las diferentes aplicaciones de Programación Genética puede encontrarse en (Martinez & Velasquez, 2015). En cuanto a la predicción del consumo eléctrico, entre los trabajos basados en Regresión Simbólica, (Jinliang Yin et al., 2008) utiliza la variante Gene Expression Programming, pero sólo tienen en cuenta las variables consumo eléctrico y temperatura como entradas. Por otro lado, (Dimoulkas et al., 2018) utiliza Programación Genética sólo para seleccionar variables de entrada, mientras que el proceso de predicción se realiza con redes neuronales. Entonces, no se tiene un modelo matemático explícito que pueda ser analizado posteriormente.

El consumo de energía se ve influenciado por componentes aleatorios y está compuesto por la contribución de muchos consumidores individuales. La Programación Genética, al igual que otras técnicas basadas en Inteligencia Artificial, son las más adecuadas para dar una respuesta en estos casos (Jimenez et al., 2017). En este trabajo exploramos el uso de una variante lineal de Programación genética llamada *Multi-Expression Programming* (MEP) para Regresión Simbólica, aplicando esta técnica en el problema de predicción del consumo eléctrico a corto plazo, un día hacia adelante. En nuestro caso, se realizará la predicción de los valores de la variable corriente eléctrica de fase de la Subestación Transformadora, ya que se dispone de mediciones de esta variable para validar los resultados. Los datos utilizados fueron provistos por la empresa de distribución de energía de Tucumán, Argentina.

El resto del documento está dividido de la siguiente manera. En primer lugar, se brinda una descripción técnica del algoritmo MEP y se describen los datos utilizados. Luego, se muestran los resultados de las pruebas de regresión simbólica con MEP en funciones de prueba y en series de tiempo de corriente eléctrica. En último lugar se presentan las conclusiones del trabajo.

PROGRAMACIÓN GENÉTICA LINEAL PARA PREDICCIÓN

La Regresión Simbólica es un problema que es abordado comúnmente con Programación Genética (Dracopoulos & Kent, 1997; McConaghy et al., 2010; McPhee et al., 2008). A diferencia de las técnicas tradicionales, GP proporciona una fórmula matemática que, aunque pueda ser difícil de entender a priori, es susceptible de interpretación y análisis posterior. Esta característica es especialmente útil en Física y Economía, donde puede ayudar a proporcionar nuevas interpretaciones, y en medicina, donde el profesional debe tener una interpretación clara de los modelos obtenidos. Tradicionalmente, los programas producidos con GP eran expresados como uno o más árboles que debían ser interpretados (Tree-Based Genetic Programming). Sin embargo, existen otros tipos de arquitecturas de GP que representan a los programas de manera diferente. Gene Expression Programming o GEP (Ferreira, 2006) utiliza una codificación especial que permite representar árboles en forma lineal. Linear Genetic Programming o Linear-GP (Brameier & Banzhaf, 2007) representa un programa mediante un conjunto de instrucciones atómicas, que se ejecutan secuencialmente.

Diversas variantes lineales de GP fueron propuestas, algunas de ellas son Simple

Revista Tecnología y Ciencia
DOI:https://doi.org/10.33414/rtyc.39.85-102.2020 ISSN 1666-6933

Linear Genetic Programming o LGP (Brameier & Banzhaf, 2007), Grammatical Evolution o GE (O'Neill & Ryan, 2001), Cartesian Genetic Programming (Miller & Thomson, 2000) y MulțiExpression Programming o MEP (Mihai Oltean & Dumitrescu, 2002). Todas ellas tienen una característica en común: los individuos son representados por entidades lineales (cadenas de texto) que son codificadas y expresadas como entidades no lineales (árboles). La variante MEP en particular presenta muchas ventajas. En primer lugar, codifica múltiples soluciones en un mismo cromosoma, lo que permite explorar eficientemente grandes zonas del espacio de búsqueda. Además, esta arquitectura no requiere utilizar intérpretes, ganando velocidad en el cálculo (McConaghy et al., 2010; Mihai Oltean & Grosan, 2003). En un trabajo previo (Mihai Oltean & Grosan, 2003), se hizo una comparación del rendimiento de las variantes lineales de GP: MEP, GEP, GE y LGP sobre un conjunto de problemas de prueba bien conocidos, conocidos con el nombre Benchmarks. MEP superó en la mayoría de los casos a las otras técnicas, seguido por LGP. Por este motivo se elige esta variante para el desarrollo de este trabajo.

Multi-Expression Programming

El algoritmo MEP comienza creando una población aleatoria de individuos. Los siguientes pasos se repiten hasta que se alcanza un número determinado de generaciones. Dos individuos padres son seleccionados mediante un procedimiento de selección. Los padres se recombinan para obtener dos descendientes. Estos nuevos individuos hijos son considerados para el proceso de mutación. La mejor descendencia reemplazará al peor individuo en la población actual, siempre que la misma tenga mejor fitness. El pseudocódigo completo del algoritmo MEP se muestra en la Figura 1.

Algoritmo MEP Estándar

Crear la población inicial aleatoriamente $P(0)$
for $t = 1$ **to** $Total\_Generaciones$ **do**
  for $k = 1$ **to** $|P(t)|/2$ **do**
    $p_1 = \text{Selecccionar}(P(t))$; *// seleccionar el primer padre*
    $p_2 = \text{Selecccionar}(P(t))$; *// seleccionar el segundo padre*
    $c_1, c_2 = \text{Cruzar}(p_1, p_2)$; *// aplicar cruzamiento*
    $c_1 = \text{Mutar}(c_1)$; *// aplicar mutación*
    $c_1 = \text{Mutar}(c_2)$; *// aplicar mutación*
    $c = \text{Minimo}(c_1, c_2)$; *// elegir al mejor hijo*
    if Fitness(c) < fitness del peor individuo en $P(t)$ **then**
      Reemplazar el peor individuo con c;
    end if
  end for
end for

Figura 1. Pseudocódigo del algoritmo MEP estándar.

Revista Tecnología y Ciencia
DOI:https://doi.org/10.33414/rtyc.39.85-102.2020 - ISSN 1666-6933

Codificación

Cada individuo, al momento de ser evaluado, debe traducirse como un programa de computadora, el cual puede ser directamente compilado o interpretado según corresponda. La representación de estos programas en MEP es independiente de cualquier lenguaje de programación. El número de genes por cromosoma es constante e igual para todos los individuos de la población. Cada gen contiene una tupla que puede ser transformada en una expresión matemática. Para la generación de expresiones se define un conjunto de primitivas que incluye:

- Conjunto de Terminales, que incluye variables de entrada (ejemplos: *x, y, z, u, v,* etc.) y constantes (ejemplos: 1, -2, 3, π, e, etc.).
- Conjunto de Funciones, que incluye operadores aritméticos (ejemplos: +, -, ×, /, etc.) y llamado a funciones conocidas (ejemplos: *cos, sin, tan, div, log, pow, mod, abs,* etc.).

En la Tabla I se muestra ejemplos de las representaciones de las expresiones en un cromosoma, su traducción fenotípica en una instrucción de código y la expresión matemática resultante de ejecutar el código hasta esa línea. En nuestro caso de aplicación en predicción de la variable eléctrica "corriente", en el conjunto de terminales se incluyen tanto variables climáticas, eléctricas y temporales (relacionadas al tiempo), como la temperatura, humedad, la corriente eléctrica de momentos anteriores, día del año, etc. En el ejemplo de la Tabla 1, la variable x podría estar representando la variable temperatura, y la expresión resultante se utiliza para predecir la corriente eléctrica (variable que se desea predecir).

Gen del cromosoma	Instrucción de Código	Expresión resultante	Gen del cromosoma
0	x	r[0] = x	
1	3	r[1] = 3	3
2	mul, 1, 0	r[2] = r[1] * r[0]	3x
3	cos, 2	r[3] = cos(r[2])	cos(3x)
4	add, 2	r[4] = r[0]+r[3]	x+cos(3x)

Tabla 1. Representaciones de las expresiones en un cromosoma en MEP

El primer gen del cromosoma debe contener un terminal. Los siguientes genes pueden contener un terminal o una función. En este último caso, el gen es representado por una tupla cuyo primer elemento es la función y los elementos siguientes son los argumentos definidos por su aridad (cantidad de parámetros que toma la función). Cada uno de estos argumentos es un valor entero que sirve de referencia a una expresión que se encuentra contenida en un gen con un índice menor al del gen actual. En la Ecuación 1 se muestra un ejemplo de un gen que contiene una función con aridad 2. Es decir, cuando un operador tiene argumentos, puede referenciar a expresiones de genes anteriores.

$$gene\_i = (function, j, k) \quad , con \quad j, k < i \tag{1}$$

Esta codificación tiene ventajas sobre la codificación tradicional de cromosoma de solución única, especialmente cuando la complejidad de la expresión buscada es desconocida. Además, la complejidad máxima de las expresiones está dada por la cantidad de genes, que es fija e igual para todos los individuos de la población. Por esa razón, no se presenta la

Revista Tecnología y Ciencia
DOI:https://doi.org/10.33414/rtyc.39.85-102.2020 ISSN 1666-6933

tendencia de los individuos a incrementar desmedidamente en tamaño sin mejorar la calidad (fenómeno conocido como *bloat*). La expresión que representa a un individuo no necesariamente usa todos los genes del cromosoma, pero la presencia de los mismos introduce diversidad en la población y pueden llegar a generar mejores soluciones en generaciones posteriores tras una sucesión de cruzamientos y mutaciones.

Función objetivo

En MEP cada gen contiene una expresión que representa una solución candidata, por lo que se debe evaluar el fitness de cada una de ellas. Luego, la expresión con mejor fitness es elegida para representar al individuo. En el caso de un problema de regresión simbólica, se parte de un conjunto de datos de entrenamiento. El mismo está formado por n valores de entrada para las k variables de entrada involucradas (matriz de tamaño $n \times k$), y los valores de salida esperada (vector W con n elementos). Dado un cromosoma de longitud m, se evalúa cada expresión del cromosoma con las variables de entrada, y se obtiene una matriz O de tamaño $n \times m$ con los valores de salida. El fitness de una expresión utiliza la métrica el error absoluto, dada por la Ecuación 2, donde $O_{k,i}$ es el valor de salida obtenido por la expresión $_i$ para la muestra de datos k, y W_k es el valor de salida esperado para la misma muestra de datos.

$$f\left(E_i\right) = \sum_{k=1}^{n} \left| O_{k,i} - W_k \right| \quad (2)$$

$$f\left(C\right) = min_i f\left(E_i\right) \quad (3)$$

Manejo de excepciones

Generalmente, cualquier algoritmo de Programación Genética genera siempre programas sintácticamente correctos. Sin embargo, durante la ejecución pueden ocurrir excepciones que interrumpen el flujo normal de ejecución del programa debido a errores de dominio matemático (por ejemplo, la división por cero). En MEP, estas excepciones solo pueden ocurrir en los genes que contienen una función. En estos casos particulares, el algoritmo muta el gen que genera esta excepción, cambiando su valor por un símbolo terminal aleatorio.

Operadores Genéticos

Para evolucionar las expresiones, MEP utiliza operadores genéticos convencionales. En el caso de la selección, se utiliza el operador de selección por torneos. Por otro lado, para el cruzamiento de individuos, se utiliza el operador de corte en un punto o el operador de cruzamiento uniforme. El único operador genético específico que usa MEP es el de mutación. Si el gen elegido para mutación es el primero, el contenido del mismo será modificado por un terminal aleatorio dentro del conjunto de terminales. Cualquier otro gen distinto del primero que sea elegido para mutación será reemplazado por un terminal o una función con las referencias a sus argumentos, respetando la regla de que los argumentos referenciados tienen que ser de genes anteriores al gen actual. Estas restricciones permiten que siempre se obtengan programas sintácticamente correctos.

Software existente

Para el desarrollo de este trabajo optamos por utilizar una implementación existente del

Revista Tecnología y Ciencia
DOI:https://doi.org/10.33414/rtyc.39.85-102.2020 - ISSN 1666-6933

algoritmo MEP, llamada *MEPx*. El software *MEPx* fue desarrollado por mismo creador de MEP (Mihai Oltean & Dumitrescu, 2002). Es una aplicación de escritorio, multiplataforma y con interfaz gráfica, pero no es un software libre. Internamente utiliza la librería *mepxlib*, que sí es software libre y puede descargarse en forma gratuita. Al estar escrita en lenguaje C y hacer uso del multiprocesamiento, las ejecuciones son eficientes y rápidas.

MEPx ofrece muchas opciones para parametrizar y llevar a cabo correctamente regresiones con Multi-Expression Programming estándar. A pesar de las limitaciones que posee, entre las cuales podemos nombrar la falta de personalización en los operadores utilizados, resulta adecuado para las pruebas que se proponen en los objetivos de este trabajo.

RESULTADOS Y DISCUSIÓN

El conjunto de datos de variables eléctricas fue provisto por la Empresa de Distribución Eléctrica de Tucumán S.A. (EDET S.A.). En este trabajo nos limitamos a utilizar las mediciones de corriente eléctrica medidos en la red de distribución de baja tensión, que corresponde a las subestaciones transformadoras de 315 kVA de potencia. Este es el nivel más bajo de agregación antes de llegar al de los clientes domiciliarios. Contamos con mediciones de corriente tomadas cada 15 minutos a lo largo de dos meses, Noviembre-2014 y Diciembre-2016.

En cuanto a los datos climáticos utilizados, fueron adquiridos por estaciones meteorológicas cercanas pertenecientes a la Estación Experimental Agroindustrial Obispo Colombres (EEAOC). Entre todas las variables disponibles, en este trabajo utilizaremos solamente la temperatura y la humedad relativa, por ser las más influyentes en la demanda eléctrica de la provincia (Jimenez et al., 2017). Con el objetivo de evaluar estadísticamente el desempeño de los modelos implementados, se analiza el error a través de distintas métricas de la Tabla II, comparando los valores de corriente eléctrica calculada (Y_i) con la corriente eléctrica medida (T_i).

Métrica	Ecuación	Interpretación
Root Mean Squared Error	$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(Y_i - T_i)^2}{n}}$	**Está expresado en la misma unidad de medida que la variable a predecir, facilita su interpretación.**
Mean Bias Error	$MBE = \sum_{i=1}^{n} \frac{Y_i - T_i}{n}$	**Permite saber si existe una subestimación o una sobreestimación, analizando su signo**
Coeficiente de Correlación Lineal de Pearson	$R = \frac{n\sum Y_i T_i - \sum Y_i \sum T_i}{\sqrt{n\sum Y_i^2 - (\sum Y_i)^2}\sqrt{n\sum T_i^2 - (\sum T_i)^2}}$	**Ayuda a determinar el grado con que los datos siguen la tendencia general del modelo.**
Mean Absolute Error	$MAE = \frac{1}{n}\sum_{i=1}^{n}\lvert Y_i - T_i\rvert$	**Puntuación lineal, por lo que todas las diferencias individuales se ponderan por igual en el promedio.**
Mean Absolute Percentage Error	$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left\lvert\frac{Y_i - T_i}{T_i}\right\rvert$	**Se expresa como un porcentaje, lo que facilita su interpretación a diferencia del MAE.**

Tabla 2. Métricas de error utilizadas para evaluar resultados de predicción.

Revista Tecnología y Ciencia
DOI:https://doi.org/10.33414/rtyc.39.85-102.2020 ISSN 1666-6933

Ajuste de parámetros

Se realizaron pruebas preliminares con funciones Benchmark con el propósito de probar el software y determinar los valores más adecuados para los parámetros involucrados. Como punto de partida se intenta encontrar los parámetros necesarios para obtener una solución correcta para la función Rastrigin, un Benchmark bien conocido para problemas de optimización y regresión simbólica (Elli et al., 2014). Ésta es definida por la Ecuación 4, donde x es la única variable de entrada de la función. La metodología a seguir consiste en ejecutar el programa con los parámetros por defecto, comparar el rendimiento de los operadores de cruzamiento *One-Cutting Crossover y Uniform Crossover,* para determinar cuál de ellos permite generar mejores modelos, evaluar si es necesario un incremento de la cantidad de generaciones y por último probar con otras funciones de prueba empleando los mismos parámetros como base. Esto no garantizará éxito en cada problema, sino que simplemente se usará como criterio común para evaluar el rendimiento de la herramienta.

$$f(x) = 0.95 * x^2 - 5.5cos(\pi * x), -8 < x < 8 \quad (4)$$

Para realizar el experimento se generó un conjunto de datos de entrenamiento de 500 puntos de la función Rastrigin. Se realizaron 25 ejecuciones en total para cada operador de cruzamiento, usando los parámetros de la Tabla III. Se tomó el error MAE del mejor individuo de cada ejecución como métrica de error.

Parámetro	Valor	Parámetro	Valor	Parámetro	Valor
Conjunto de terminales	x; 2.7183; 3.1416; 5; 7	Probabilidad de cruzamiento	0.9	Probabilidad de constantes	0.3
Conjunto de funciones	+; -; *; /; cos; sin	Probabilidad de mutación	0.01	Cantidad de Generaciones	100
Cantidad de Subpoblaciones	4	Tamaño del torneo	2	Cantidad de hilos	4
Tamaño de subpoblación	300	Probabilidad de operadores	0.4	Cantidad de ejecuciones	5
Longitud del cromosoma	50	Probabilidad de variables	0.3	Métrica de error	MAE

Tabla 3. Parámetros utilizados para las pruebas con Benchmarks

En la Figura 2 se muestra un histograma con los resultados obtenidos con cada operador de cruzamiento. A partir del histograma, puede observarse que el operador Uniform Crossover tiene una tendencia a obtener mayor cantidad de soluciones con error bajo. Además, Uniform Crossover tiene un promedio de MAE por ejecución inferior, demostrando ser una mejor opción para la regresión de la función Rastrigin bajo estas condiciones.

Revista Tecnología y Ciencia
DOI:https://doi.org/10.33414/rtyc.39.85-102.2020 - ISSN 1666-6933

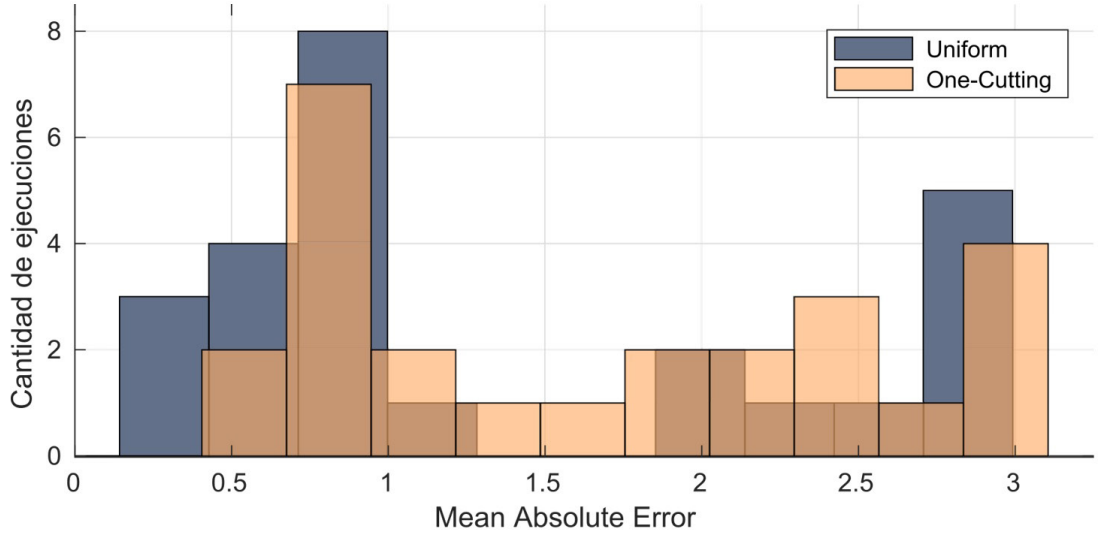

Figura 2. Histograma de error absoluto promedio de las mejores soluciones en 25 ejecuciones.

A continuación, se determina la cantidad de generaciones adecuada para el problema. Para ello, nos valemos de la información de mejor fitness y fitness promedio en cada generación a lo largo de la ejecución. Al menos 20 de las 25 ejecuciones no presentaban signos de convergencia, con una notable diferencia entre el mejor individuo y el promedio. En la Figura 3 se muestra las gráfica de la evolución para dos ejecuciones particulares. Lamentablemente, el software no brinda información sobre los ejes de la figura, pero se pueden utilizar para comparar dos ejecuciones diferentes. La Figura 3(a) corresponde a una ejecución con 100 generaciones, donde hay una tendencia a seguir mejorando si es que hubiese más generaciones. Esto se comprueba aumentando el número de generaciones a 500, como se observa en la Figura 3(b).

De la experiencia del uso de MEP para predicción de series de tiempo, se puede concluir que el parámetro más importante a definir es la longitud del cromosoma. Con este parámetro fijamos cuán compleja puede llegar a ser nuestra solución. Dependiendo del caso, será necesaria más o menos complejidad para representar los datos. Sin embargo, esto trae aparejado dos problemas. En primer lugar, los modelos matemáticos resultantes pueden llegar a ser muy complejos, difíciles de leer y analizar. En segundo lugar, puede que se produzca un sobre-ajuste (también llamado *overfitting*), donde la expresión resultante modela el ruido presente en los datos.

Por otro lado, una vez seleccionada la longitud del cromosoma, es recomendable hacer ejecuciones del algoritmo con poblaciones pequeñas, por ejemplo 100, e ir incrementando según sea necesario. De este modo, se aumenta la diversidad y se previene que el algoritmo converja prematuramente, ayudándonos para esto del gráfico de evolución de la población. Por último, la diversidad no solo depende de la cantidad de individuos, sino también de la longitud de los cromosomas, ya que cada gen de un cromosoma codifica una solución candidata. Entonces, para reducir el procesamiento requerido, hay que encontrar un balance entre la longitud del cromosoma y el tamaño de la población.

Revista Tecnología y Ciencia
DOI:https://doi.org/10.33414/rtyc.39.85-102.2020 ISSN 1666-6933

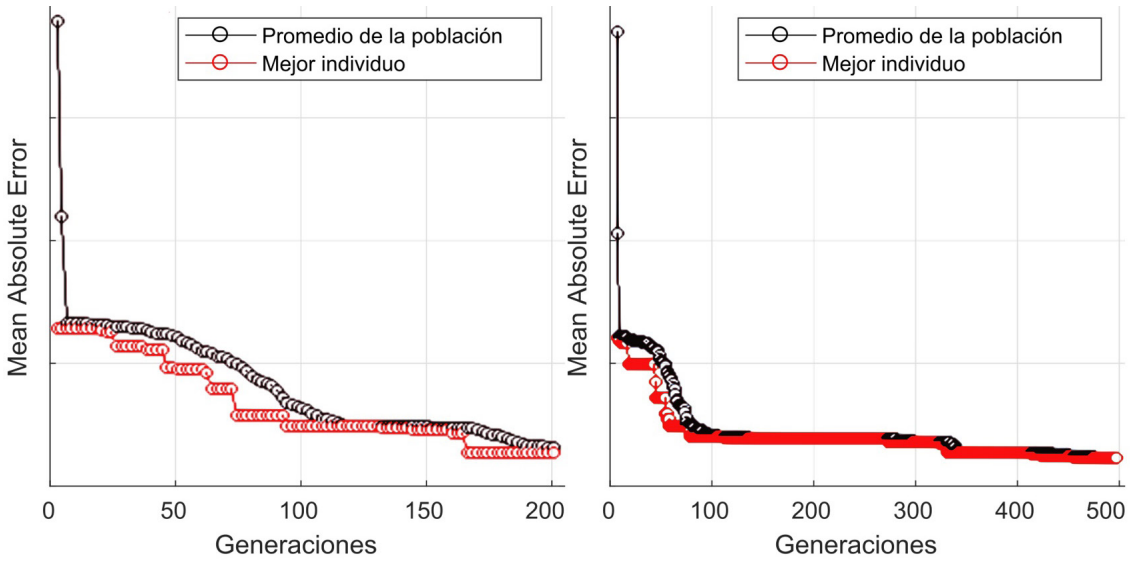

Figura 3. Curvas del mejor fitness y el fitness promedio de la población en cada generación.

Predicción de la corriente eléctrica

En todos los modelos obtenidos se utilizaron los parámetros de la Tabla III con algunas modificaciones. Por un lado, la probabilidad de mutación se incrementa a 0.05, para asegurar que el algoritmo tenga buena exploración del espacio de búsqueda. Por otro lado, se agrega el valor 0.5 a la lista de constantes. Tanto la lista de operadores disponibles se mantiene igual como el resto de los parámetros se mantiene, ya que dieron buenos resultados en pruebas preliminares.

En cuanto a las variables de entrada utilizadas para modelar la corriente eléctrica, las principales son la temperatura (*t*) y humedad relativa (*h*), porque son las que presentan mayor correlación con respecto a la demanda de electricidad. Además, se consideran dos variables temporales: la hora del día, representada como el seno y el coseno del ángulo de rotación terrestre (*art*), y el mes del año, expresado en decimales representado por el seno y coseno del ángulo de traslación terrestre (*att*). Esta manera de representar las variables temporales ayuda al modelo a capturar mejor la estacionalidad, aportando de manera más adecuada información sobre componentes cíclicos en el consumo eléctrico. Por otro lado, también se hace uso de variables con retraso (llamadas también variables *lag*), es decir, variables adicionales que tienen valores de instantes pasados de las variables de entrada. Las variables con retraso permiten incluir y utilizar información del pasado para ayudar a mejorar los pronósticos de la corriente eléctrica. El uso de estas variables es muy común en la literatura (Jimenez et al., 2017; Lizondo et al., 2015).

Para el conjunto de entrenamiento se utilizaron los datos del mes de noviembre y parte del mes de diciembre, y se validaron los modelos obtenidos con datos de la segunda mitad del mes de diciembre, continuando con la serie temporal (estos últimos datos no intervienen en ninguna etapa del procedimiento de obtención del modelo). Es importante destacar que los datos de entrenamiento y validación corresponden a la misma época del año. Esto es importante ya que la demanda de electricidad y su dependencia con las otras variables

Revista Tecnología y Ciencia
DOI:https://doi.org/10.33414/rtyc.39.85-102.2020 - ISSN 1666-6933

varían de acuerdo a la época del año (por ejemplo, es de esperarse que en épocas de calor la temperatura y la humedad tengan una gran influencia).

A continuación se describen los modelos obtenidos con MEP, detallando los parámetros utilizados y el nivel de error logrado por cada uno de ellos

Modelo 1: Además de las variables de corriente eléctrica (*c*), temperatura ambiente (*t*) y los ángulos de rotación terrestre (*sa, ca*), se utilizan variables *lag* correspondiente a 1, 3 y 24 horas anteriores tanto de la corriente (*c*1, *c*3, *c*24) como de temperatura (*t*1, *t*24). La longitud del cromosoma es de 150 para alcanzar la complejidad necesaria. Como el incremento de la longitud del cromosoma en MEP implica un aumento de la diversidad, no es necesario tener una población tan grande, por lo tanto se fija cada subpoblación a 3000 (3 subpoblaciones). La prueba se realizó con un máximo de 1500 generaciones.

Se consiguió un modelo que logra seguir muy bien la tendencia de los datos, con un MAPE final de 5.45% y un R^2 de 0.89, que se ve reflejado en las curvas de corriente de la Figura 4. Observando la evolución de la población en la Figura 5, puede observarse que los parámetros ingresados fueron ajustados correctamente, no hubo convergencia prematura y en las últimas generaciones no se ve una tendencia de que la población vaya a mejorar mucho más. La fórmula matemática resultante es la siguiente, donde se considera d_{cs}=ca-sa y s_{cs}=-ca+sa sólo para simplificar la expresión:

$$M_1 = c + 3\times t - 3\times t_1 + s_{cs} \times \left(d_{cs} + d_{cs} \times s_{cs} + s_{cs} + s_{cs}^2 - d_{cs} \times sa - d_{cs} \times s_{cs} \times sa\right)^4 -$$
$$s_{cs} \times (t - t_1) - (ca \times (sa + d_{cs} \times s_{cs}) + sa) \times (d_{cs} + d_{cs} \times s_{cs} \times (d_{cs} + d_{cs} \times s_{cs} + s_{cs} + s_{cs}^2 -$$
$$d_{cs} \times sa - d_{cs} \times s_{cs} \times sa)^4 + d_{cs} \times s_{cs} + 5) + (d_{cs} + ij + s_{cs} + s_{cs}^2 - d_{cs} \times sa - d_{cs} \times s_{cs} \times sa)^2$$

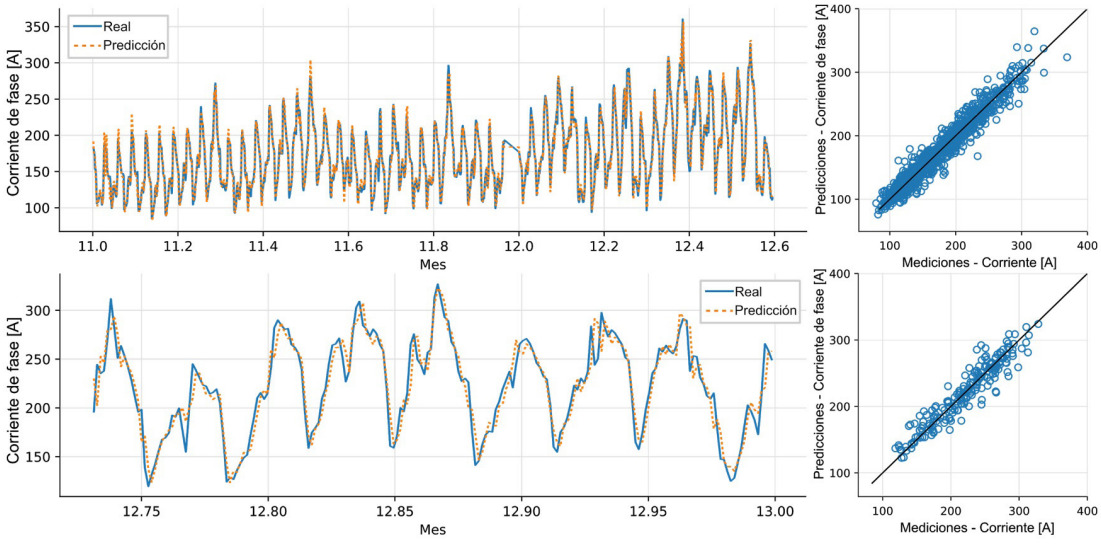

Figura 4. Curvas de la corriente eléctrica medida de una de las fases de la subestación transformadora y las predicciones realizadas con MEP (izquierda), y gráficos de valores reales vs estimados (derecha).

Revista Tecnología y Ciencia
DOI:https://doi.org/10.33414/rtyc.39.85-102.2020 ISSN 1666-6933

Modelo 2: Para obtener el modelo anterior, la variable de entrada temperatura utilizada corresponde a mediciones reales (no son predicciones). En la actualidad, existen modelos meteorológicos que nos permiten tener una predicción de la temperatura con bajo nivel de error en el corto plazo. Haciendo uso de esta ventaja, se utilizarán los valores de temperatura del momento que se desea predecir, suponiendo que éstos corresponden a predicciones muy precisas de temperatura. Otra diferencia respecto al caso anterior es que para este modelo no usaremos variables *lag* inferiores a las 24 horas, ya que demostraron no ser significativas.

Tomamos inicialmente una longitud de 25 para los cromosomas, subpoblaciones de 3000 individuos y 700 generaciones. El resto de los parámetros permanecerán sin cambios. Con este modelo se logró un MAPE de 8,0 y un R^2 de 0.77, los cuales son ligeramente peores que en el modelo anterior. Presenta algunas subestimaciones que se deben a las variables no incluidas en estas pruebas y si en la anterior. Sin embargo, la adición de la variable temperatura con los valores predichos pudo contrarrestar la falta de estas variables. El modelo resultante está dado por la siguiente expresión:

$$M_2 = 20 \times ca + 0.6624 \times c_{24} + 3 \times t_{24} - 6.2832 + \left(0.0455 \times c_{24} - 3.1416\right) \times \left(t - t_{24}\right)$$

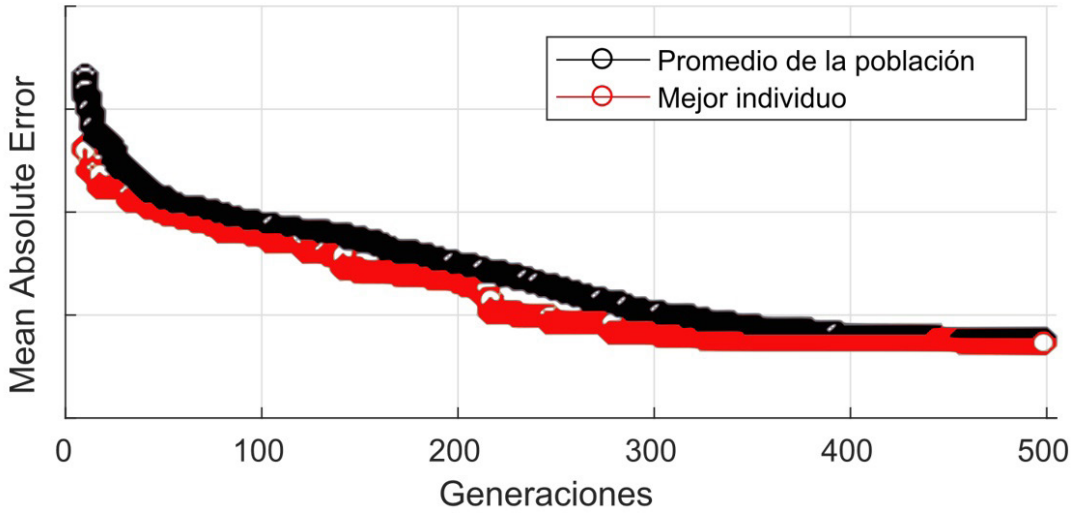

Figura 5. Valores de mejor fitness y fitness promedio en cada generación en una ejecución particular del algoritmo MEP.

Modelo 3: Por último, se realiza una prueba incluyendo la variable humedad (*h*), manteniendo los parámetros con los mismos valores utilizados previamente. El resultado de esta prueba es el mejor modelo logrado, con un valor MAPE de 5.54 y R^2 de 0.9 en el conjunto de validación. Curiosamente la variable humedad no fue incluida. Si bien el análisis cualitativo del modelo obtenido no es parte de los objetivos de este trabajo, se puede presuponer que el algoritmo encontró una manera de compensar la información que aporta la variable humedad utilizando solamente la variable temperatura. Este modelo está dado por la siguiente expresión simplificada.

Revista Tecnología y Ciencia
DOI:https://doi.org/10.33414/rtyc.39.85-102.2020 - ISSN 1666-6933

$$M_3 = c_1 + c_1 / (sa - 7.5) + d_{cs} + t + sa^2 / (2 \times ca \times sa + sa - 0.5) - (sa - 0.5) \times (10.8732 \times ca^2 \times sa^2 + 4.7183 \times ca) / (2 \times s_{cs})$$

El listado final de variables es el siguiente:

- *t*: Temperatura (valores predichos).
- *h*: Humedad relativa.
- *sa, ca*: sen y coseno del ángulo de rotación terrestre.
- *c*1, *c*3, *c*24: variable *lag* de 1, 3, y 24 horas de la corriente eléctrica.
- *t*1, *t*24: variable *lag* de la temperatura de 1 y 24 horas.
- *h*1: variable *lag* de la humedad relativa de 1 hora.

En la Tabla IV se muestran los niveles de error logrados por cada modelo, tanto en el conjunto de entrenamiento como en el de validación.

N°	Datos	MAE	MAPE	MBE	RMSE	R^2
1	Entrenamiento	8.6	5.15	-0.15	11.8	0.95
	Validación	11.6	5.45	0.08	15.6	0.89
2	Entrenamiento	14.9	9.02	0.02	19.4	0.85
	Validación	17.4	8.00	-4.46	22.8	0.77
3	Entrenamiento	8.5	5.13	-0.52	11.7	0.95
	Validación	12.0	5.54	2.45	15.3	0.90

Tabla 4. Nivel de error de las predicciones obtenidas con cada modelo encontrado con programación genética.

CONCLUSIONES

En este trabajo se analizó la variante de programación genética Multi-Expression Programming para su uso en la predicción a corto plazo de la corriente eléctrica, utilizando software existente. Se utilizaron datos reales provenientes de Subestaciones Transformadores ubicadas en la provincia de Tucumán, Argentina.

Utilizando diferentes conjuntos de variables de entrada se consiguieron tres diferentes modelos que permiten predecir la corriente eléctrica con similar nivel de error, logrando un error RMSE de 15.3 A en el mejor caso. A pesar de que se incluyó la variable humedad, el modelo resultante no la consideró en la expresión final, utilizando en su lugar información de las otras variables disponibles. Sin embargo, es importante incluir la humedad en el conjunto de entrada, para permitir ser utilizada en modelos obtenidos para otras Subestaciones Transformadoras.

El nivel de error logrado es ligeramente inferior a los obtenidos con otras técnicas reportadas en un trabajo previo (Jimenez et al., 2017). Utilizando Redes Neuronales Artificiales se obtuvo un error RMSE de 16.1 A en datos de Subestaciones de la misma provincia. Por lo tanto, concluimos que MEP es una técnica con mucha potencialidad para la estimación o

Revista Tecnología y Ciencia
DOI:https://doi.org/10.33414/rtyc.39.85-102.2020 ISSN 1666-6933

predicción de series temporales.

Finalmente, al utilizar el software existente no se pudo implementar variaciones en el algoritmo para mejorar los resultados. Se propone como trabajo futuro elaborar una propia implementación de Multi-Expression Programming que emplee operadores genéticos más adecuados, y explorar otras arquitecturas de algoritmos genéticos.

AGRADECIMIENTOS

Agradecemos a la Empresa de Distribución de Energía de Tucumán S.A. (EDET S.A.) y a la Estación Experimental Agroindustrial Obispo Colombres (EEAOC) por proporcionar los datos necesarios para la elaboración de este trabajo.

Revista Tecnología y Ciencia
DOI:https://doi.org/10.33414/rtyc.39.85-102.2020 - ISSN 1666-6933

REFERENCIAS

Abd Jalill, N. A., Ahmad, M. H., & Mohamed, N. (2013). Electricity Load Demand Forecasting Using Exponential Smoothing *Methods. World Applied Sciences Journal*, 22(11), 1540-1543. https://doi.org/10.5829/idosi.wasj.2013.22.11.2891

Aguilar-Rivera, R., Valenzuela-Rendón, M., & Rodríguez-Ortiz, J. (2015). Genetic algorithms and Darwinian approaches in financial applications: A survey. *Expert Systems with Applications,* 42(21), 7684–7697. https://doi.org/10.1016/j.eswa.2015.06.001

Ahmad, W. M. A. W. (2012). Forecasting short term load demand using multilayer feed-forward (MLFF) neural network model. *Applied Mathematical Sciences*, 6(108), 5359–5368.

Amral, N., Ozveren, C. S., & King, D. (2007). Short term load forecasting using Multiple Linear Regression. *Universities Power Engineering Conference, 2007. UPEC 2007. 42nd International*, 1192-1198. https://doi.org/10.1109/UPEC.2007.4469121

Arora, Y., Singhal, A., & Bansal, A. (2014). A Study of Applications of RBF Network. *International Journal of Computer Applications, 94*(2). https://doi.org/10.5120/16315-5553

Borges, C. E., Peña, A., & Penya, Y. K. (2013). On the influence of surrounding load demand to improve primary substation STLF. *IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society*, 8166-8171. https:/v/doi.org/10.1109/IECON.2013.6700499

Brameier, M. F., & Banzhaf, W. (2007). *Linear genetic programming*. Springer Science & Business Media.

Ceperic, E., Ceperic, V., & Baric, A. (2013). A Strategy for Short-Term Load Forecasting by Support Vector Regression Machines. *IEEE Transactions on Power Systems, 28*(4), 4356–4364. https://doi.org/10.1109/TPWRS.2013.2269803

Dimoulkas, I., Herre, L., Khastieva, D., Nycander, E., Amelin, M., & Mazidi, P. (2018). A Hybrid Model Based on Symbolic Regression and Neural Networks for Electricity Load Forecasting. *2018 15th International Conference on the European Energy Market (EEM)*, 1-5.

Dracopoulos, D. C., & Kent, S. (1997). Genetic programming for prediction and control. *Neural Computing & Applications, 6*(4), 214–228. https://doi.org/10.1007/BF01501508

Elli, S., Jimenez, V. A., Will, A., & Rodriguez, S. (2014). Optimización de Constantes Numéricas en Regresión Simbólica utilizando un Framework de Tree-Based Genetic Programming. *Actas del 2° Congreso Nacional de Ingeniería Informática/Sistemas de Información,* 114-124.

Ferreira, C. (2006). *Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence (Studies in Computational Intelligence).* Springer-Verlag New York, Inc.

Gandomi, A. H., Alavi, A. H., & Ryan, C. (2015). *Handbook of genetic programming applications.* Springer.

Government, U. S. (2011). *Nist Framework and Roadmap for Smart Grid Interoperability Standards, Release 1 0.* General Books.

Jimenez, V. A., Elli, S., Will, A., & Rodriguez, S. (2015). Optimización de Constantes Numéricas en Tree-Based Genetic Programming. *Tecnología y Ciencia,* (27), 184-196.

Jimenez, V. A., Lizondo, D., Will, A., & Rodriguez, S. (2017). Short-Term Load Forecasting

Revista Tecnología y Ciencia
DOI:https://doi.org/10.33414/rtyc.39.85-102.2020 ISSN 1666-6933

for Low Voltage Distribution Lines in Tucumán, Argentina. *5to Congreso Nacional de Ingeniería Informática / Sistemas de Información (CoNaIISI 2017)*, 940-949.

Jinliang Yin, Limin Huo, Lirui Guo, & Jie Hu. (2008). Short-term load forecasting based on improved gene expression programming. *2008 7th World Congress on Intelligent Control and Automation*, 5647-5650.

Kamboj, R., & Avtar, R. (2013). Electric Load Forecasting Using Different Techniques in BPN. I*nternational Journal of Advanced Engineering Technology,* IV, 75-78.

Koza, J. R. (1994). Genetic programming as a means for programming computers by natural selection. *Statistics and Computing,* 4(2), 87–112. https://doi.org/10.1007/BF00175355

La Cava, W., Spector, L., Danai, K., & Lackner, M. (2014). Evolving differential equations with developmental linear genetic programming and epigenetic hill climbing. *Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation*, 141–142.

Lee, C.-M., & Ko, C.-N. (2009). Time series prediction using RBF neural networks with a nonlinear time-varying evolution PSO algorithm. *Neurocomputing, 73(*1–3), 449-460. https://doi.org/10.1016/j.neucom.2009.07.005

Lee, C.-M., & Ko, C.-N. (2011). Short-term load forecasting using lifting scheme and ARIMA models. *Expert Systems with Applications*, 38(5), 5902-5911. https://doi.org/10.1016/j.eswa.2010.11.033

Lizondo, D. F., Jimenez, V. A., Villacis Postigo, F., Will, A., & Rodríguez, S. (2015). Análisis de Variables Temporales para la Predicción del Consumo Eléctrico. *Revista Técnica Energía*, 11, 5-12.

Martinez, C. A., & Velasquez, J. D. (2015). Conceptual Developments in Genetic Programming for Time Series Forecasting. *IEEE Latin America Transactions, 13*(8), 2728-2733. https://doi.org/10.1109/TLA.2015.7332156

McConaghy, T., Vladislavleva, E., & Riolo, R. (2010). Genetic programming theory and practice 2010: An introduction. *Genetic Programming Theory and Practice VIII,* 8.

McPhee, N. F., Poli, R., & Langdon, W. B. (2008). *Field guide to genetic programming.*

Miller, J. F., & Thomson, P. (2000). Cartesian genetic programming. E*uropean Conference on Genetic Programming*, 121–132.

Nataraja, C., Gorawar, M. B., Shilpa, G. N., & Shri Harsha, J. (2012). S*hort Term Load Forecasting Using Time Series Analysis: A Case Study for Karnataka, India. 1*(2). https://doi.org/10.1109/TAPENERGY.2015.7229635

Oltean, Mihai, & Dumitrescu, D. (2002). Multi Expression Programming, Technical Report. *UBB-01-2002, Babes-Bolyai University, Cluj-Napoca, Romania.*

Oltean, Mihai, & Grosan, C. (2003). A comparison of several linear genetic programming techniques. *Complex Systems, 14*(4), 285–314.

O'Neill, M., & Ryan, C. (2001). Grammatical evolution. *IEEE Transactions on Evolutionary Computation, 5*(4), 349–358.

Oplatkova, Z. K., Senkerik, R., & Viktorin, A. (2018). Differential Evolution and Analytic

Revista Tecnología y Ciencia
DOI:https://doi.org/10.33414/rtyc.39.85-102.2020 - ISSN 1666-6933

Programming in the case of Trigonometric Identities Discovery. *2018 25th International Conference on Systems, Signals and Image Processing (IWSSIP)*, 1–5.

Palacio, D. R. (2001). Simple model for load forecast weather sensitive. E*lectricity Distribution, 2001. Part 1: Contributions. CIRED. 16th International Conference and Exhibition on (IEE Conf. Publ No. 482)*, 4, 3–pp.

Rothe, M. J. P., Wadhwani, A. K., Dr., & Wadhwani, M. S., Dr. (2009). Short Term Load Forecasting Using Multi Parameter Regression. *ArXiv e-prints.*

Salimi-beni, A., Farrokhzad, D., Fotuhi-Firuzabad, M., & Alemohammad, S. J. (2006). A New Approach to Determine Base, Intermediate and Peak-Demand in an Electric Power System. 2*006 International Conference on Power System Technology, 1-5*. https://doi.org/10.1109/ICPST.2006.321928

Sandineni, S., & Boehm, R. (2012). Measurements and simulations for peak electrical load reduction in cooling dominated climate. *Energy,* (37), 689-697. https://doi.org/10.1016/J.ENERGY.2011.10.026

Sevlian, R., & Rajagopal, R. (2014). Short Term Electricity Load Forecasting on Varying Levels of Aggregation. *arXiv preprint arXiv:1404.0058*. http://arxiv.org/abs/1404.0058

Troiano, L., Vaccaro, A., & Vitelli, M. C. (2016). On-line smart grids optimization by case-based reasoning on big data. *2016 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS)*, 1–6. https://doi.org/10.1109/EESMS.2016.7504842

Tuaimah, F. M., & Abass, H. M. A. (2014). Short-Term Electrical Load Forecasting for Iraqi Power System based on Multiple Linear Regression Method. I*nternational Journal of Computer Applications, 100*(1), 41-45. https://doi.org/10.5120/17492-8011