International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 21 (2018) pp. 14876-14885

Machine Learning Techniques and Smart Grid Applications: A Review

William Sanchez-Huertas[1], Víctor Gómez[1], Cesar Hernández[1*]

[1] *Universidad Distrital Francisco José de Caldas, Technological Faculty, Bogotá, Colombia.*

[*]*Corresponding author, [1*]Orcid: 0000-0001-9409-8341*

Abstract:

Machine learning tools are computations useful in data processing to find hidden patterns or the prediction of results. The objective of this document is to compare the most used techniques of machine learning, such as: Vector Machine Support, Descriptive Discriminant Analysis, Decision Trees and Neural Networks, in Smart Grid applications. To this end, an investigation is carried out in relevant publications of the current literature.

Keywords: Algorithms, distributed energy resources, machine learning, smart grid

INTRODUCTION

The steady evolution of computational methods, specifically in data management and analysis has enabled several machine learning techniques to be implemented in smart grid applications. According to [1], Opower, a leading company in energy information, drives quantifiable savings through energy efficiency based on information.

One of the main challenges of using smart grids is the capacity to manage electric and communication networks, where the amount of real-time information (power quality, price, energy demand, etc.) is linked to the nodes of the system. Hence, the focus lies on creating artificial intelligence models that can make decisions, even in uncertain conditions (see Figure 1) [2].

In general, definitions of artificial intelligence are related to the development of methods and algorithms that allow computers to carry out processes and make decisions in a similar way as humans do [3].

In section 2, different machine learning techniques are discussed. They can be implemented in solving problems associated with the integration and management of Smart Grids. In section 3, some applications regarding the security and efficiency of electric networks based on machine learning are shown. In section 4, a comparative evaluation of characteristics is established in order to choose an algorithm. Finally, section 5 includes a series of conclusions.

MACHINE LEARNING TECHNIQUES

Machine learning is a data analysis technique in which computers are taught to make decisions based on experience. This learning process uses calculation methods to 'learn'. The algorithms improve their performance adaptively as the number of available samples increases. With the increase in the amount of Big Data, machine learning has become a crucial technique to solve problems. It involves two methods: supervised learning, which trains a model with previously known input and output data so it can predict future outputs and unsupervised learning, which finds hidden patterns and intrinsic structures within the input data [4].

In Figure 2, machine learning techniques are classified into supervised and unsupervised:

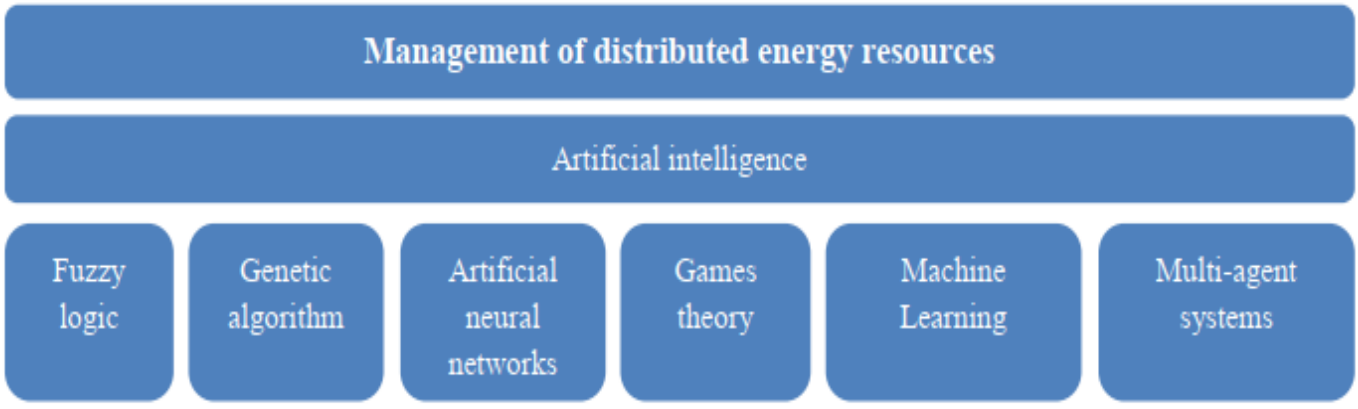

Figure 1. Artificial Intelligence System for DMS.
Source [2]

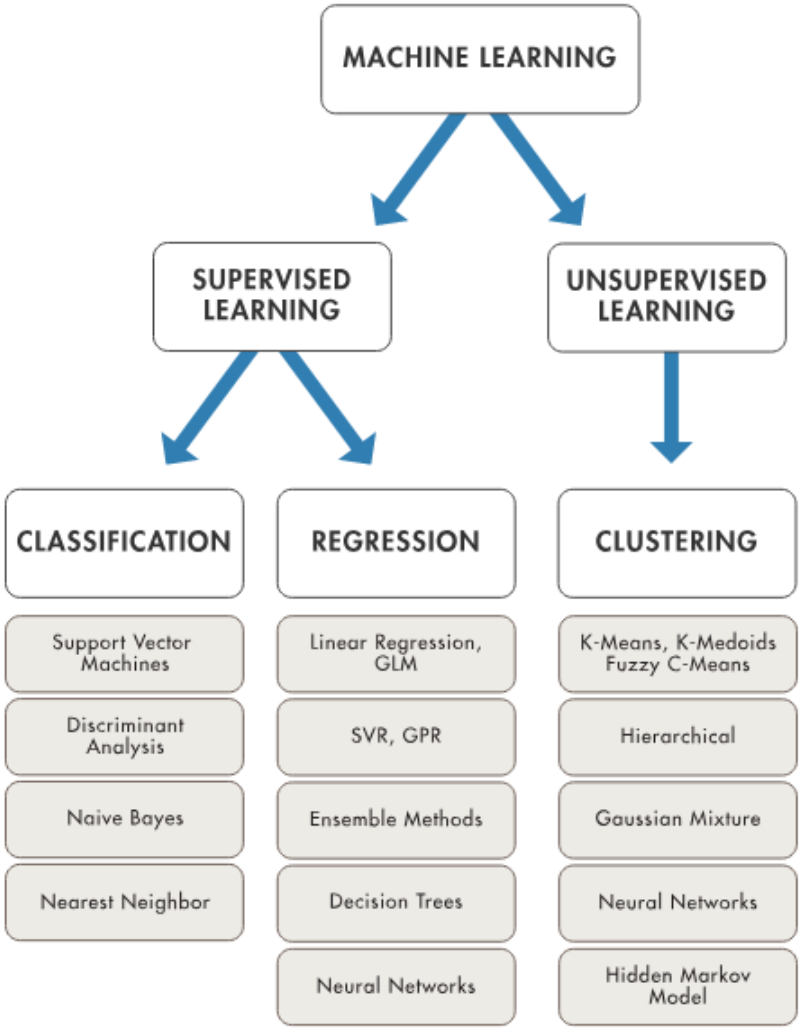

Figure 2. Machine Learning techniques. Source [4]

The most relevant algorithms are analyzed in current literature.

Support Vector Machines (SVM)

Support vector machines (SVM) come from the work on stochastic learning theory and were introduced in the 90s by Vapnik and his colleagues [5]. Although SVM were originally conceived for solving binary classification problems, they are currently used to solve other types of problems (regression, grouping, multiclass classification). There are several fields in which SVM have worked successfully such as computer vision, character recognition, text and hypertext categorization, protein classification, natural language processing and time series analysis. In fact, since their introduction, they have earned a well-deserved recognition thanks to their solid theoretical foundation [6].

According to [7], "a SVM maps the entry points into a characteristic space of higher dimension (i.e. if the entry points are in R^2 then they are mapped by the SVM to R^3) and finds a hyperplane that separates them and maximizes the margin *m* between the classes in said space" (Figure 3).

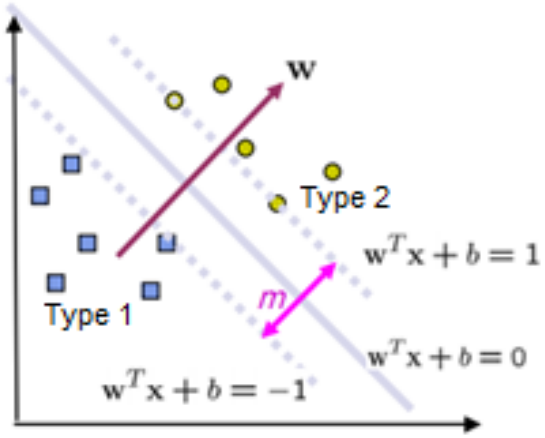

Figure 3. The decision frontier must far from the data. Source [7]

The proposed cases for classification problems are described.

Linearly separable case

Each training point $X_i \in R^N$ belongs to one of two classes and has been labeled as $y_i \in \{-1, 1\}$ para i = 1, . . ., l. In most cases, the search of an adequate hyperplane in an input space is too restrictive to be of practical use. A solution to this situation consists on mapping the input space into a space of characteristics with a higher dimension and seeking the optimal hyperplane. Let z = ϕ(x) be the notation of the corresponding vector in the space of characteristics with a mapping ϕ of N ℜ into a space of characteristics Z [7]. The desired hyperplane is described by (1).

$$w.z + b = 0 \qquad (1)$$

defined by the pair (w, b) such that the point x_i can be separated according to the function described in (2).

$$f(x_i) = \text{sign}(w.z_i + b) = \begin{cases} 1 & y_i = 1 \\ -1 & y_i = -1 \end{cases} \qquad (2)$$

Where w ∈ Z and b ∈ R. More precisely, the set S is said to be linearly separable if there is a pair (w, b) that satisfies (3).

$$\begin{cases} (w.z_i + b) \geq 1, & y_i = 1 \\ (w.z_i + b) \leq -1, & y_i = -1 \end{cases} \quad i = 1, \ldots, l \qquad (3)$$

In Figure 4, the linearly separable case is shown.

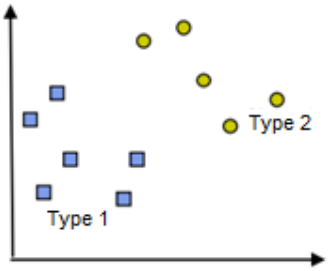

Figure 4. Linearly separable case. Source [7]

Non-linearly separable case

If the set S is not linearly separable, violations to the classification must be allowed in the formulation of the SVM [7]. Figure 5 describes the non-linearly separable case. To handle non-linearly separable data, the previous analysis can be generalized by introducing some non-negative variables $\xi_i \geq 0$ so that (3) is modified and turned into (4).

$$yi\ (w\ .zi\ +\ b)\ \geq\ 1\ -\ \xi \qquad (4)$$

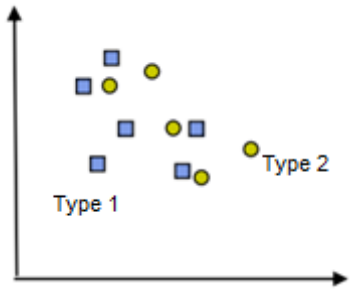

Figure 5. Linearly separable case. Source [7]

The $\xi_i \neq 0$ in (4) are those for which the term X_i does not satisfy (3). Then, the term $\sum_{i=1}^{l} \xi_i$ can be taken as some sort of measurement of the classification error.

Descriptive Discriminant Analysis (DDA)

The Discriminant Analysis is a statistical multivariable technique whose purpose is to analyze if there are significant differences between groups of objects regarding a set of variables measured on said objects. In the case that they exist, their nature must be explained and hence facilitate procedures for systematical classification of new observations of an unknown origin in one of the analyzed groups [8].

A problem of discriminant analysis is presented when it is necessary to explain a qualitative variable based on a certain number of quantitative variables called explained variables or predictors. DDA can be understood as a set of methods and statistical procedures oriented to two objectives that may be complementary [9]: (1) Determine if the observed variables can distinguish (discriminate) the *r* groups a priori. This objective has descriptive nature and is related to the principal components analysis. It is obvious at this point that importance is given to the construction of bidimensional representations of the individuals, variables and groups a priori. (2) Build classification rules (decisional rules) to assign each new object to one of the groups a priori. This objective is decisional and is linked to probabilistic methods. For this approach, both the construction of decision rules and the procedures for subsequent evaluation are crucial.

Calculation of discriminant functions

The main purpose of DDA consists on determining *m* variables z^1, ..., z^m called discriminant functions, which have (and are determined by) the following properties:

- Each $z^j \in R^p$ is a linear combination of the *p* variables (see equation (5))

$$z^j = \sum_{s=1}^{p} a_{js} x^s = X a_j \qquad (5)$$

where $\alpha_j = (\alpha_{j1}, ..., \alpha_{jp})^t$; j = 1, . . ., m.

- The variables are D orthonormal, i.e., they are not correlated and have a variance of 1.
- The values of each variable for the individuals of the same group must be as close as possible. This means that the term intra (zj → intraclass variance) must be minimized.
- The values of each variable z^j for the individuals belonging of different classes must be as different as possible. This implies that the term inter (z^j → interclass variance) must be maximized.

Algorithm for the computational implementation of DDA [9]

In this implementation, the same notations introduced in the previous text are used and it is assumed that all individuals have equal weight: $p_i = 1/n$, $i = 1, 2, \ldots, n$. The number of groups a priori must be higher than 2, i.e., $r > 2$. For the calculation of

the cosines of individuals and the barycenters, the identities described by equations (6) and (7):

$$x_i V^{-1} x_i^t = \sum_{h=1}^{t} [coord_h(x_i)]^2 \quad (6)$$

$$g_l V^{-1} g_l^t = \sum_{h=1}^{t} [coord_h(g_l)]^2 \quad (7)$$

- Centering and reduction operations:
  - For *j=1, ..., p* equations (8) and (9) are computed.

$$x^j = \frac{1}{n} \sum_{i=1}^{n} x_{ij} \quad (8)$$

$$\sigma_j^2 = \frac{1}{n} \sum_{i=1}^{n} x_{ij}^2 - (x^j)^2 \quad (9)$$

  - For *i=1, ..., n* and *j = 1, ..., p* the input x_{ij} from *X* is replaced by $x_{ij-x}{}^{-j}$ for centering, and by $\frac{xij-x-j}{\sigma_j}$ for reduction.
- Barycenters of the groups for *l = 1, ..., r*; j = 1, ..., p, compute $g_{il} = \frac{1}{nl} \sum i \in E_l \, x_{ij}$, with $n_l = |E_l|$
- Calculate $V = \frac{1}{n} X^t X = (v_{ij})_{p \, x \, p}$, where $v_{ij} = \frac{1}{n} \sum_{S=1}^{n} x_{si} x_{sj}$ for *i,j = 1, ..., p*.
- Calculate V^{-1}
- *C* is the matrix defined as $C = C_g^t D_q^{\frac{1}{2}}, p \, x \, q$. If $r \leq p$, calculate $A = C^t V^{-1} C$, its non-null values, orthonormal eigenvectors I_r- and the corresponding βj.
  - For l, k =1, …, r calculate equation (10)

$$(C^t V^{-1} C)_{lk} = \frac{\sqrt{n_l n_k}}{n} \sum_{b=1}^{p} g_{bk} \left[\sum_{f=1}^{p} g_{fl} \, t_{ft}\right] \quad (10)$$

  - Calculate the eigenvalues $\lambda_1 \geq \ldots \geq \lambda_t \geq 0$ of A and the eigenvectors corresponding to $\omega_1, \ldots, \omega_t$, with I_r – orthonormalized.
  - Calculate $\beta_j = \frac{c\omega_j}{\sqrt{\lambda_j}}$, which implies calculating (11) for *j = 1, ..., t* y *s = 1, ..., p*.

$$\beta_{sj} = \frac{1}{\sqrt{n\lambda_j}} \sum_{k=1}^{r} \sqrt{n_k} g_{sk} \omega_{kj} \quad (11)$$

- If *r > p*, calculate A = HBH as well as its non-null values, its orthonormalized eigenvectors I_p- and the corresponding β_h.
  - Calculate B and H: for *i, j = 1, ..., p* compute equations (12) and (13):

$$B_{ij} = \frac{1}{n} \sum_{s=1}^{r} n_s \, g_{is} g_{js} \quad (12)$$

$$H_{ij} = \sum_{k=1}^{p} \frac{1}{\sqrt{\mu k}} u_{ik} u_{kj} \quad (13)$$

  - For *i, j = 1, ..., p*, calculate HB and HBH according to equations (14) and (15):

$$(HB)_{ij} = \sum_{S=1}^{p} H_{is} B_{sj} \quad (14)$$

$$(HBH)_{ij} = \sum_{S=1}^{p} (HB)_{is} H_{sj} \quad (15)$$

  - Calculate the eigenvalues $\lambda_1 \geq \ldots \geq \lambda_t \geq 0$ from A and the corresponding eigenvectors $\omega_1, \ldots, \omega_t$, with I_p – orthonormalized.
  - Calculate $\beta_j = H^{-1} \omega j$; and for *j = 1, ..., t* y *s = 1, ..., p* calculate (16)

$$B_{sj} = \sum_{k=1}^{p} \omega_{kj} \left[\sum_{h=1}^{p} \sqrt{\mu h} \, u_{sh} \, u_{hk}\right] \quad (16)$$

- Calculation of coordinates
  - Barycenters, individuals and variables
- Calculation of square cosines:
  - Individuals: For *s = 1, ..., t* and *i = 1, ..., n*, calculate (17):

$$cos^2(i,s) = \frac{[coord_s(x_i)]^2}{\sum_{h=1}^{t} [coord_h(x_i)]^2} \quad (17)$$

  - Barycenters: For *s = 1, ..., t* and *l = 1, ..., r*, calculate (18):

$$cos^2(l,s) = \frac{[coord_s(g_l)]^2}{\sum_{h=1}^{t} [coord_h(g_l)]^2} \quad (18)$$

Decision Tree (DT)

A decision tree (DT) is a representation of a multivariate function which can be used in practical everyday applications through the development of modern computers. The interest for their practical use had its origin in the needs of Social Sciences with the work of Son Quist-Morgan (1964), the AID (Automatic Interaction Detection) software. This was one of the first methods of data adjustment in classification trees. DT transcended from just being an illustrative representation in decision-making courses to becoming a useful and easy-to-use tool [10].

In [11], decision trees are also defined as "graphical representations of possible solutions to decision-making processes under certain conditions". It is one of the most commonly used supervised learning algorithms in machine learning and can perform classification or regression tasks. The understanding of its functioning tends to be simple and robust at the same time.

To generate the optimal tree, assess each subdivision among all possible trees and determine the root node and subsequent nodes, the algorithm must measure the predictions obtained, assess them to compare them and choose the best one. In order

to measure and assess, it uses diverse functions being *Index Gini* and *Information Gain* the most well-known. The latter uses the term known as entropy. The division of nodes will be repeated until the maximum possible depth of the tree is reached or the nodes are limited to a minimal quantity of samples in each leaf [11].

Index of Gini

The index of Gini is a criterion based on impurities that measures the convergence among probability distributions of the values of targeted attribute [12] .

The index of Gini is obtained through equation (19).

$$\text{Gini}\,(y, S) = 1 - \sum \left(\frac{|\sigma_y - cjS|}{|S|}\right)^2 \quad (19)$$

Information Gain

Categorical attributes such as (men / women) are established in order to estimate the information brought by each attribute based on information theory. Entropy is used to measure the uncertainty randomness of the random value of variable *X*.

A decision tree is formed by a set of decision nodes (internal nodes) and response nodes (leaves):

- A decision node is linked to one of the attributes and has one or more branches springing out of it. Each branch represents the possible values of the corresponding attribute. The decision node can be seen as a question being asked to the analyzed example and, depending on the given answer, the flow will take one of the outgoing branches.
- A response node is linked to the desired classification and delivers the tree's decision regarding the input example [13].

Building decision trees

A decision tree partitions the space of predictor variables into a set of hyper-rectangles and each one adjusts a simple model, often in the form of a constant such as $y = c$ where y is the answer variable [14].

The construction of the decision tree is based on four elements [14]: (1) a set of Q binary questions in the form $\{x \in A?\}$ is a subset of the sample space of variable *x*; (2) the method used to partition the nodes; (3) the strategy required to stop the tree's growth; and (4) the allocation of each terminal node to a value of the answer variable (regression) or to a class (classification).

The main differences between the algorithms for building trees are found in the rule to partition nodes, the strategy to prune trees and the treatment of missing values.

Example of a Decision Tree

The following example proposes a match of golf. Table 1 shows the information gathered regarding the weather's condition and the possibility to play golf or not [14].

Table 1. Compilation regarding the meteorological conditions and the possibility of playing golf or not playing. Adapted from [15]

Predictors				Target
Outlook	**Temp**	**Humidity**	**Windy**	**Play Golf**
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

In Figure 6, the decision tree shows a possible mechanism learned to make decisions in this classification task. Additionally, the decision nodes (in dark blue) and the answer nodes (in light orange) make up the decision tree along with the labels in the output branches (in light blue).

Figure 6. Decision tree to play golf. Source [15]

Neural networks

An Artificial Neural Network (ANN) is a mathematical model inspired in the biological behavior of neurons and how the structure of the brain is organized. The brain can be considered as a highly complex system, where it is estimated that approximately 100.000 million neurons in the brain cortex form a framework of over 500 billion of neural connections. One neuron can have up to 100.000 connections even though the average varies between 5.000 and 10.000 connections [16].

Advantages of neural networks

Due to their constitution and foundation, artificial neural networks have a large number of characteristics similar to those of the brain. For instance, they are capable of learning from experience, generalizing previous cases for newer cases, abstracting essential characteristics based on inputs that represent irrelevant information, among other things; this technology offers many advantages and is being applied in multiple areas [17]. Some of the advantages include [17]:

- **Adaptive Learning**: Ability to learn to perform tasks based on training or an initial experience.
- **Self-Organization:** A neural network can create its own organization or representation of the information it receives during a learning stage.
- **Tolerance to failures:** The partial destruction of a network leads to a degradation of its structure; However, some capabilities of the network can be maintained, while important damages are taken.
- **Operation in real-time:** Neural computations can be performed in parallel. For this purpose, machines with specialized hardware are designed and produced.
- **Easy insertion within the existing technology:** Custom chips can be used to improve the capacity of neural networks in certain areas. This will facilitate the modular integration into existing systems.

Basic elements of a neural network

Figure 10 shows the schematic of a neural network, which is made of interconnected neurons arranged into three layers. The data enters through the input layer, going through the hidden layers and finally exiting through the output layer.

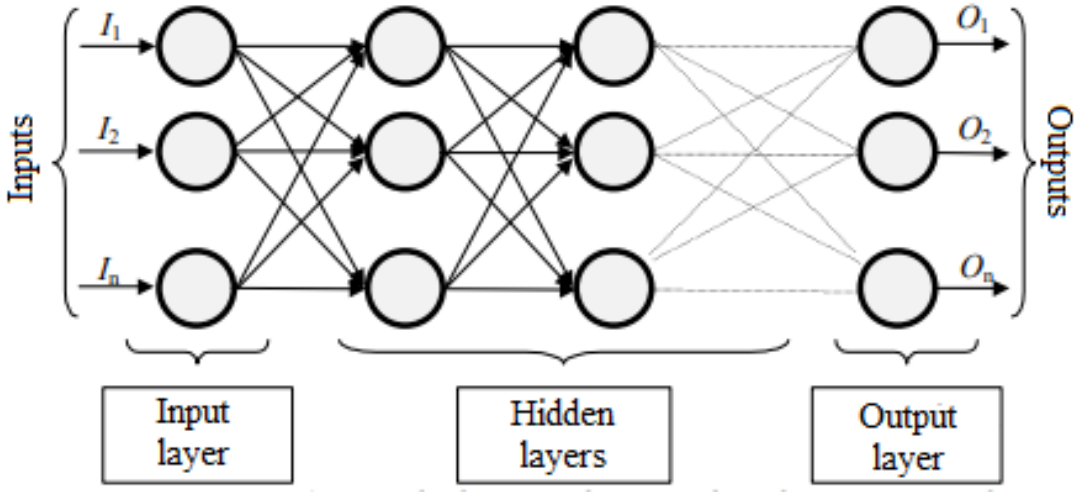

Figure 7. Example of a totally connected neural network. Source:[17]

There are two important learning methods that can be defined:

- Supervised Learning (Error-correction learning, Reinforced Learning and Stochastic Learning)
- Unsupervised Learning (Hebbian Learning, Competitive Learning and Comparative Learning)

APPLICATION OF MACHINE LEARNING TECHNIQUES

The implementation of machine learning techniques can involve the stages of the integrated electric network with renewable energies such as the prediction of energy on a short and medium term basis. This section discusses some of the applications of machine learning algorithms.

Singh in [18] proposes a genetic algorithm (GA) to take on a planning problem in which the best location and dimensioning of a DG unit needs to be optimized inside a radial network of 12 buses. The objective is to maximize the benefits of distribution network operators (DNO), quantified in the savings of electricity costs due to the use of DG. The author uses a GA based on Darwin's natural selection, in which the sample population keeps iteratively changing (there are several places and sizes of DG). Genetic operators such as mutation and mixing are adopted until the best solution can be reached in terms of size and location [19].

In [20], a model is proposed to forecast electric energy prices in Colombia by using artificial neural networks. Two network structures are used taking as input a series of daily prices in the first one and a series of prices plus the average level of the reservoirs in the second one. The results are compared in a Generalized Auto-Regressive Conditional Heteroscedastic (GARCH) model offering advantages within the sampling time but a better performance of neural networks outside the sample. The historical data was obtained from the XM Company belonging to the ISA group, with a total of 120 days of training and 31 days of the following month to check the forecast.

In [21], a sequential machine learning algorithm with an incremental solver of minimum squares is compared with a SVM regarding the management of a building's smart grid. The document states a machine learning platform with the mentioned algorithms and applied to smart networks based on SG. It can analyze the movement of the occupants, offer short term forecasts on energy as in [22] and allocate renewable energy resources. In first place, the profile of the occupant is captured by the indoors positioning system in real time with Wi-Fi-based data analysis. The energy profile is extracted through a real time measuring system that analyzes the electric loads. Afterwards, the occupant's profile and the 24-hour energy profile are fused with the prediction using an online-distributed machine learning strategy that updates its database in real time. Based on the forecasted profile of the occupant and his energy consumption, the solar energy source is allocated in the additional electric network in order to reduce the maximum demand of the main electric network.

In [23], Saeed Ahmed, Youngdoo Lee and Insoo Koo propose a supervised machine learning scheme to detect the hidden cyberattack in the estimated measuring data that are gathered through a smart grid's communication network. It is a mechanism for the selection of characteristics based on machine learning for the detection of CDC (cyber deception coverage) attacks that infringe the communication network data within the SG. Hence, a genetic algorithm was used to select the distinctive and discriminatory characteristics. The optimal characteristics that are chosen are used as an input for the SVM to detect incorrect data which are inserted in the dataset by cyber-pirates that have previous knowledge on the network topology. The SVM automatically learns the decision limit that can be attained by the maximum geometric deviation between the unverified and compromised data points. Then, it classifies the testing data as either compromised or uncompromised.

According to [24], security is one of the main challenges of new projects in networks and communications. The lack of perfect scalability, programmability and remote management has led to the rise of new network paradigms such as SDN which grants the mentioned features along with other perks. SDN gives flexible control of computer networks by organizing consumers in the network data plane via a centralized controller. However, progress is still slow as far as achieving a consistent and reliable security solution. Networks have progressively evolved as well as the attacks that they can suffer. The main concern is that the traditional security protocols do not grant the adequate protection to said networks. Recent advances in artificial intelligence, machine learning, deep learning and macro-data offer opportunities to tackle the challenges on network security and their correct behavior. The tests are carried out with different learning techniques such as genetic algorithms, SVM, cluster center + K-NN and ANN have shown detection accuracy of over 99%.

In [25], the SG is a large scale network and is an integral part of the Internet of Things (IoT). For a more effective Big Data analysis, reliable solutions for IoT networks are being designed so that many decisions can be made in real time at the edge of the network, close to where the data are generated. Gaussian functions are widely applied in the field of statistical machine learning, pattern recognition, adaptive algorithms for function approximation, etc.

COMPARATIVE ASSESSMENT OF MACHINE LEARNING TECHNIQUES

In this section, a comparative table is shown regarding the characteristics of the most used algorithms in machine learning. The table is based on articles [26], [27] and [28] that includes the following considerations to choosing an algorithm.

Precision (C1): The result does not need always to be as precise as possible. In some cases, an approximation is enough depending on the desired use. Hence, processing time can be considerably reduced when using approximations. Another advantage of these methods lies on avoiding over-adjustment.

Training time (C2): The number of minutes or hours required to train a model varies a lot depending on the algorithm. Often, training time depends on accuracy and vice versa.

Linearity (C3): Many machine learning algorithms use linearity. Linear classification algorithms such as logistic regression and support vector machines suppose that classes must be separated by a straight line (or their counterpart in higher dimension environments) and that the data follows a straight-lined trend. Although these suppositions are not incorrect for some problems, they can reduce accuracy. Some methods are implemented in Azure Machine Learning.

Number of parameters (C4): Parameters are buttons activated by scientist when initializing an algorithm. They are numbers that affect the behavior of the algorithm, such as tolerance to error or the number of iterations or even variants in the behavior of the algorithm. The training time and accuracy of the

algorithm can be very sensitive and require only the correct configuration. Usually, algorithms with a large number of parameters require more tests and possible errors to find a good balance.

Frequency of use (C5): According to the research presented in [27], it is the frequency of the techniques appearing as the main technique in articles published between 2010 and 2011 in databases such as Science Direct, Scopus, IEEEXplore Digital Library and Web of Knowledge.

Level of combination with other Machine Learning techniques (C6): The research in [27] states that it is the frequency in which machine learning techniques are combined in articles published between 2010 and 2011 in databases such as Science Direct, Scopus, IEEEXplore Digital Library and Web of Knowledge.

Sensibility (C7): Based on [28], sensibility is the probability of classifying an element as the positive class and that really belongs to the positive class. Sensibility is calculated as the proportion of positive cases that had a positive result in the prediction process.

Accuracy (C8): This variable corresponds to the rate of accurate predictions, i.e., the number of elements classified correctly.

In Table 2, a comparative is presented based on [26],[27] and [28] in which the previous competences of each algorithm are assessed. Since this assessment is qualitative, for the *precision* criterion they are labeled as *"Excellent (E)"*, *"Good (G)"* or *"Deficient (D)"*. In the *training time* criterion, the labels are "Brief (B)" or "Moderate (M)". For the *linearity* criterion, it is stated whether it uses or not a linear behavior. The *parameters* criterion indicates the number of optimal variables for the algorithm. For the "*frequency of use criterion as main technique", "level of combination with other ML techniques*" *"Sensibility"* and *"Accuracy*", qualitative labels are used: *"Very High"*, *"High (H)"*, *"Medium (M)"* and *"Low (L)"*. Finally, in the *notes* section, a relevant and brief observation is made on the algorithm.

Table 2. Comparative table of ML algorithms. Adapted from [26],[27], [28].

Machine Learning Technique	C1	C2	C3	C4	C5	C6	C7	C8	Notes
Multi-class classification									
Decision Tree	E	M	NO	6	L	M	H	H	
Decision Jungle	E	M	NO	6	L	M	M	M	Low memory usage
Boosted Decision Tree	E	M	NO	6	L	M	M	M	Large memory usage
Neural Network	E	-	NO	9	L	M	M	L	Further customizations are possible
Support Vector Machines	-	M	SI	5	H	VH	M	M	Useful for large sets of characteristics
Targeted Local Support Vector Machines	B	-	NO	8	H	VH	M	M	Useful for large sets of characteristics
Bayes point machine	-	B	SI	3	M	H	L	M	
Regression									
Bayesian Linear Regression	-	M	SI	2	M	H	L	M	
Decision Forest	E	M	NO	6	L	M	H	H	
Boosted Decision Tree	E	M	NO	5	L	M	M	M	Large memory usage
Fast Forest Quantile	E	M	NO	9	-	M	M	M	Distributions instead of point-based predictions
Neural Network	E	-	NO	9	L	M	M	L	Further customizations are possible
Anomaly detection									
Support Vector Machines	B	M	NO	2	H	VH	M	M	Especially useful for large sets of characteristics
K-Means	-	B	SI	4	M	M	M	M	A cluster-based grouping algorithm

CONCLUSIONS

The development of computational systems has opened the door to a world of opportunities for machine learning applications through algorithms or hybrid methods that improve efficiency and are becoming progressively more powerful and capable of processing large amounts of information. The information that needs to be processed is ever-increasing, requires more accuracy, lower training times and faster response times.

The migration of the electric sector towards smart grids demands the continuous development of machine learning techniques since their implementation can harmoniously integrate all the components used which grants reliability in smart electric systems as well as guaranteeing a service of quality, efficiency and continuity.

ACKNOWLEDGEMENTS

The authors of this article wish to acknowledge Colciencias and the Universidad Distrital Francisco José de Caldas for funding resources to develop this research.

REFERENCES

[1] Business Wire, "Opower Announces 10th Utility Customer in Minnesota; Saves Residents More Than $6 Million on Energy Bills," *Business Wire*, 2011. [Online]. Available: https://www.businesswire.com/news/home/20110817005814/en/Opower-Announces-10th-Utility-Customer-Minnesota-Saves. [Accessed: 20-Jul-2018].

[2] V. A. Gómez, C. Hernandez, and E. Rivas, "Management of Distributed Energetic Resources," vol. 12, no. 20, pp. 9506–9514, 2017.

[3] R. P. Ino, "¿Que es Inteligencia Artificial?"

[4] MathWorks, "Aprendizaje automático: Tres cosas que es necesario saber - MATLAB & Simulink." [Online]. Available: https://la.mathworks.com/discovery/machine-learning.html. [Accessed: 23-Jun-2018].

[5] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A Training Algorithm for Optimal Margin Classiiers."

[6] E. J. Carmona Suárez, "Tutorial sobre Máquinas de Vectores Soporte (SVM)."

[7] G. Betancour, "Las máquinas de soporte vectorial (SVMs)," *Sci. Tech.*, no. 27, pp. 67–72, 2005.

[8] S. De La Fuente Fernández, "Análisis Discriminante," 2011.

[9] Universidad de Costa Rica. Centro de Investigación en Matemática Pura y Aplicada., W. Castillo, and O. R. Rojas, *Análisis Discriminante Descriptivo: Teoría, Algoritmo y Software*, vol. 6, no. 1. Centro de Investigación en Matemtáica Pura y Aplicada, Universidad de Costa Rica, 2009.

[10] C. N. Bouza and A. Santiago, "La Minería De Datos: Arboles De Decisión Y Su Aplicación En Estudios Médicos," vol. 2, no. November, pp. 64–78, 2012.

[11] J. I. Bagnato, "Arbol de Decisión en Python: Clasificación y predicción.," *APRENDE MACHINE LEARNING*, 2018. [Online]. Available: http://www.aprendemachinelearning.com/arbol-de-decision-en-python-clasificacion-y-prediccion/. [Accessed: 02-Jul-2018].

[12] L. Rokach and O. Maimon, "Top-Down Induction of Decision Trees Classifiers—A Survey," *Appl. Rev.*, vol. 35, no. 4, 2005.

[13] F. Sancho Caparrini, "Aprendizaje Inductivo: Árboles de Decisión," 2018. [Online]. Available: http://www.cs.us.es/~fsancho/?e=104. [Accessed: 02-Jul-2018].

[14] E. Acuna, "Mineria de Datos Arboles de Decision."

[15] J. R. Quinlan, "Induction of Decision Trees," *Mach. Learn.*, vol. 1, pp. 81–106, 1986.

[16] F. Sancho Caparrini, "Redes Neuronales: una visión superficial," 2017. [Online]. Available: http://www.cs.us.es/~fsancho/?e=72. [Accessed: 02-Jul-2018].

[17] C. A. Ruiz, M. Susana, B. Autor, and J. D. Matich, "Universidad Tecnológica Nacional – Facultad Regional Rosario Departamento de Ingeniería Química Grupo de Investigación Aplicada a la Ingeniería Química (GIAIQ) Redes Neuronales: Conceptos Básicos y Aplicaciones," 2001.

[18] B. Biswal, D. Sattianadan, M. Sudhakaran, and S. S. Dash, "Optimum Allocation of Distributed Generation Based on Nodal Pricing for Profit and Social Welfare Maximization," *Adv. Mater. Res.*, vol. 768, pp. 364–370, Sep. 2013.

[19] V. Gomez, C. Hernandez, and E. Rivas, *Desarrollo , políticas y futuro de las microredes inteligentes,* Primera Ed. Bogotá: Universidad Distrital Francisco José de Caldas, 2018.

[20] F. Villada, D. R. Cadavid, and J. D. Molina, "Pronóstico del precio de la energía eléctrica usando redes neuronales artificiales," *Rev. Fac. Ing.*, no. 44, pp. 111–118, 2014.

[21] H. Xu, H. Huang, R. S. Khalid, and H. Yu, "Distributed machine learning based smart-grid energy management with occupant cognition," *2016 IEEE Int. Conf. Smart Grid Commun. SmartGridComm 2016*, pp. 491–496, 2016.

[22] Q. Cheng, J. Yao, H. Wu, S. Chen, C. Liu, and P. Yao, "Short-term load forecasting with weather component based on improved extreme learning machine," *2013 Chinese Autom. Congr.*, pp. 316–321, 2013.

[23] S. Ahmed, Y. Lee, S. Hyun, and I. Koo, "Feature Selection – Based Detection of Covert Cyber

Deception Assaults in Smart Grid Communications Networks Using Machine Learning," vol. 6, 2018.

[24] A. Hussein, A. Chehab, A. Kayssi, and I. H. Elhajj, "Machine Learning for Network Resilience : The Start of a Journey," pp. 59–66, 2018.

[25] E. Oyekanlu, S. Onidare, and P. Oladele, "Towards Statistical Machine Learning for Edge Analytics in Large Scale Networks : Real-Time Gaussian Function Generation with Generic DSP."

[26] MICROSOFT AZURE, "Cómo elegir algoritmos de aprendizaje automático," *Microsoft Docs*, 2017. [Online]. Available: https://docs.microsoft.com/es-es/azure/machine-learning/studio/algorithm-choice. [Accessed: 08-Jul-2018].

[27] Á. F. Godoy Viera, "Técnicas de aprendizaje de máquina utilizadas para la minería de texto," *Investig. Bibl. Arch. Bibl. e Inf.*, vol. 31, no. 71, p. 103, Mar. 2017.

[28] J. Martínez Rodríguez, "Estudio comparativo de modelos de machine learning para la detección de dianas microARN," *Publ. by Univ. Oberta Catalunya*, 2018.