UNIVERSIDADE FEDERAL DE ITAJUBÁ - UNIFEI
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

A Comparison of Machine Learning Regression models for Critical Bus Voltage and Load mapping with regards to Max Reactive Power in PV buses.

Fernando Fachini

Itajubá, 29 de janeiro de 2021

UNIVERSIDADE FEDERAL DE ITAJUBÁ - UNIFEI
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Fernando Fachini

A Comparison of Machine Learning Regression models for Critical Bus Voltage and Load mapping with regards to Max Reactive Power in PV buses.

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica como parte dos requisitos para obtenção do Título de Mestre em Ciências em Engenharia Elétrica.

Área de Concentração: Sistemas Elétricos de Potência

Orientador: Prof. Dr. Benedito Isaias Lima Fuly

29 de janeiro de 2021
Itajubá

UNIVERSIDADE FEDERAL DE ITAJUBÁ - UNIFEI
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

A Comparison of Machine Learning Regression models for Critical Bus Voltage and Load mapping with regards to Max Reactive Power in PV buses.

Fernando Fachini

Banca Examinadora:
Prof. Dr. João Alberto Passos Filho
Prof. Dr. Cairo Nascimento Junior
Prof. Dr. Giscard F. Cintra Veloso
Prof. Dr. Guilherme Sousa Bastos

Itajubá

Fernando Fachini

A Comparison of Machine Learning Regression models for Critical Bus Voltage and Load mapping with regards to Max Reactive Power in PV buses/ Fernando Fachini. – Itajubá, 29 de janeiro de 2021-

65 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Benedito Isaias Lima Fuly

Dissertação (Mestrado)
Universidade Federal de Itajubá - UNIFEI
Programa de pós-graduação em engenharia elétrica, 29 de janeiro de 2021.

1. Palavra-chave1. 2. Palavra-chave2. I. Orientador. II. Universidade xxx. III. Faculdade de xxx. IV. Título

CDU 07:181:009.3

Fernando Fachini

A Comparison of Machine Learning Regression models for Critical Bus Voltage and Load mapping with regards to Max Reactive Power in PV buses

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica como parte dos requisitos para obtenção do Título de Mestre em Ciências em Engenharia Elétrica.

Prof. Dr. Benedito Isaias Lima Fuly
Orientador

Prof. Dr. João Alberto Passos Filho

Prof. Dr. Cairo Nascimento Junior

Prof. Dr. Giscard F. Cintra Veloso

Prof. Dr. Guilherme Sousa Bastos

Itajubá
29 de janeiro de 2021

Agradecimentos

Primeiramente, quero agradecer a Deus por ter me dado saúde e força de vontade para correr atrás dos meus sonhos.

Agradeço ao meu orientador, Dr. Isaias, por ter acreditado no meu trabalho e ter me instruído da melhor forma possível.

Aos meus pais, muito obrigado por me darem todo o amor do mundo. Desejo continuar a dar-lhes orgulho do filho que criaram.

Agradeço também o meu irmão que sempre me apoia e me encoraja em todos os meus projetos e planos.

Agradeço a minha linda esposa Jéssica, que sempre esteve comigo, independentemente das circunstâncias. Ela embarcou na "loucura"planejada de morarmos nos EUA, onde hoje continuo minha formação acadêmica no doutorado. Sem ela nada seria possível.

E, por último, e não menos importante, eu gostaria de agradecer os meus amigos de república Andre, Diego e Gabriel. A vida acadêmica foi muito mais leve e divertida devido à amizade e inúmeras risadas que tivemos diariamente.

"Nobody Cares. Work Harder"
(Cameron Hanes)

Resumo

O objetivo desta dissertação de mestrado é comparar as capacidades de mapeamento de tensão e carregamento do sistema para uma variedade de algoritmos de regressão, como Adaptive Neuro-Fuzzy Inference System (ANFIS), Artificial Neural Networks (ANN), K-Nearest Neighbors (KNN), Support Vector Regression (SVR) e Decition Tree (DT). Uma matriz de sensibilidade à tensão é gerada a partir da matriz Jacobiana do fluxo de potência para um cenário de carregamento próximo ao ponto instável. A Análise de Componentes Principais (PCA) é usada para separar o sistema, próximo ao ponto crítico, a fim de agrupar os barramentos em áreas de controle de tensão coerentes. Para diferentes cenários de injeção de potência reativa, temos diferentes tensões de barramento que podem ser mapeadas pelos algoritmos de regressão mencionados acima. Os algoritmos são treinados com quantidades limitadas de dados, a fim de estabelecer uma comparação justa entre eles. O presente trabalho mostra que ANFIS e KNN têm um melhor desempenho em tensão crítica e previsão de carga quando comparados aos demais. Os sistemas de barramento acadêmico IEEE 14 e 118 são empregados com todos os seus limites considerados, portanto os resultados podem ser reproduzidos.

Palavras-chaves: Algoritmos de Regressão, ANFIS, KNN, PCA, Áreas de controle de tensão

Abstract

The aim of this master thesis is to compare voltage and system loading mapping capabilities of a variety of regression algorithms, such as Adaptive Network based Fuzzy Inference System (ANFIS), Artificial Neural Networks (ANN), K-Nearest Neighbors (KNN), Support Vector Regression (SVR), and Decision Tree (DT). A voltage sensitivity matrix is generated from the power flow Jacobian matrix for a loading scenario near the unstable point. Principal Component Analysis (PCA) is used to separate the system, close to the critical point, in order to group the buses into coherent voltage controlling areas. For different reactive power injection scenarios, we have different bus voltages that can be mapped by the aforementioned regression algorithms. The algorithms are trained with limited amounts of data, in order to establish a fair comparison between them. The present work shows that ANFIS and KNN have a better performance in critical voltage and load prediction when compared to the rest. The academic IEEE 14 and 118 bus systems are employed with all its limits considered, so the results may be reproduced.

Key-words: Regression Algorithms, ANFIS, KNN, PCA, Voltage Controlling Areas.

Lista de ilustrações

Lista de tabelas

Sumário

1 Introduction

1.1 General

According to an anual Electrical Power Research Institute (EPRI) report on power supply, it is estimated that the US has an economic loss of approximately $ 104 to $ 164 billion dollars due to power supply outages [1]. The reasons for outages can vary significantly depending on location, climate, infrastructure, etc. The most common factors that imply in outages are:

1. Ambient dependent factors such as: wind, ice storms, extreme heat, earthquakes, lightning, etc. It is not uncommon to see tree limbs interfering with low voltage distribution lines in urban areas, especially in Brazil.
2. Animals are also a major factor in short circuits. A variety of animals rely on distribution line poles for their nests, namely squirrels and birds.
3. High Power Demand, especially in extreme climate conditions where heating and cooling is at its peak.

The biggest problems occur when the system is facing high power demand, a condition where the system is stressed to a point where the boundary between normal operation and complete voltage collapse are intertwined. These power outages have the potential of disconnecting millions of civilians from the grid, an energy disruption that can cost millions of dollars in profit. Indonesia faced a serious power outage leaving roughly 100 million people without electricity for 5 hours due to deficiencies in the power generation capacity. The outage began at 10 am on August 2005 when power failed along the electrical system that connects Java, Bali, and Madura, causing outages in Java and Bali. Roughly half of the country relies on this stretch of the power grid for electricity, therefore it is important to understand what caused the abrupt loss of electricity [2]. Similarly to Indonesia, India faced what is referred to as the worst power failure up to date [3]. The power outage started at July 30th 2012, where an estimated 32 gigawatts of generation capacity went offline. The number of civilians impacted in this sudden energy loss was above 600 million people, which is close to 10% of the number of people planet's population and it lasted approximately 24 hours. According to the Power Minister Sushilkumar Shinde, electricity usage was above normal which lead to a cascading effect of generation loss, leaving millions without power.

All of the aforementioned incidents are related to voltage stability issues. These problems have caught the attention of researchers due to several voltage collapse incidents worldwide. This problem is linked to the lack of reactive power supplies in heavily stressed systems. In most cases, this is caused by system faults. Thus, voltage collapse and reactive power planning are intrinsically connected, where suitable conditions of reactive power reserves are necessary for a secure operation of the system. That being said, electric power utilities are investing in more stable grids, enhancing the stability margin as well as improving voltage stability assessment [4, 5]. A lot of good work has been done on voltage stability analysis in the last three decades, for example [6, 4, 7, 8] to name a few, and it is still a relevant topic within power systems.

1.2 Voltage Stability

The modern electric power grid, as we know, is constantly being challenged with a rising increase of load demand. Transmission systems are generally equipped with a combination of different loads (industrial, commercial, and residential) which fluctuate energy consumption throughout the day. In normal operation conditions, the system is deemed stable, and there is an equilibrium between generated and consumed power. During the year, the electrical grid faces numerous load scenarios, in which some are less drastic while other force the system to operate very close to its limit. It is in the latter condition that small changes in the load pattern may threaten the voltage stability of the system. Therefore, it is extremely important to identify all insecure operating conditions in advance for corrective measures and planning of what to do in case of system instability. This dispute is known as voltage instability and it is due to the voltage drop when the system is lacking reactive power supply [7].

There are several solutions involving voltage stability problems, mainly control changes/additions and even load shedding. Although load shedding is considered a practical solution, it is still economically expensive and does not solve the core issue. Other measures, like new control implementation and installation of new generation units are solutions for the long run and are not implemented immediately. What is valid and can be implemented in the short run, irrespective of the physical system, are better studies and computational resources to help the system operator take faster and better decisions.

Traditionally, voltage stability has been studied with the help of model based methods, and there has been a lot of debate whether quasi-static models output reliable information of the system. Although time domain simulations are capable of capturing the events in a chronological order, simulation run-time is still a major bottleneck. Electromechanical transient models also lack the ability of generating indices, which are easily done through linearized models, such as the continuation power flow [9]. With that in

mind, there is space for model free approaches in voltage stability studies.

1.3 State of the Art

The following section is a brief overview of some important papers that discuss machine learning applications in voltage stability topics.

- Malbasa et al. [10], addresses an active machine learning technique for monitoring voltage stability in transmission lines, using standard methods such as Support Vector Machine (SVM) and Artificial Neural Networks (ANN) for state estimation. Voltage stability in transmission lines are among the major challenges that system operators face during daily monitoring and control, therefore researchers have been refining predictive models for better and more robust state estimations. This is due to the fact that power systems are frequently operated near its physical limits for reasons of efficiency and economics. Model-free estimation methods, such as Machine Learning generated models, are always faced with a recurring drawback that constantly comes to play, which is data quality. Although phasor measurement unit (PMU) data is abundant nowadays, the prepossessing step in model-free models must be done correctly to avoid gruesome estimation errors. Model based methods, on the other hand are, are only prone to errors that relate to the fidelity of the model, but they are computationally expensive and can limit the operators ability to generate precise and fast decisions when needed. It was with this in mind that the authors proposed an active learning technique of iteratively building a knowledge base, for optimized size and accuracy. They relied on an interactive selection of the most representative operation points when building the training data set, making simulation and decision making efficient and fast.

- Zhou et al. [11], addresses an artificial neural network (ANN) based method in order to estimate long-term voltage stability margins. The traditional continuation power flow method (CPF), used to determine the load margin in voltage stability related problems, was proposed by Ajjarapu - Christy in paper [12] and has been widely referenced as one of the leading investigative papers in voltage stability. Similarly to the latter work, model based methods have major drawbacks, mainly the computational burden. This led the research team to propose an ANN based method that can successfully estimate the voltage stability margin for normal operation and N-1 contingency situations. Also, with PMU data being abundant, this can lead to real-time estimation of the system's state, enabling fast control actions from the system operator side. The multilayer perceptron (MLP) was the machine learning method chosen for this study, with data being generated by 3000 random power flow and load margin continuation power flow simulations. This data was then used

to train the MLP to estimate voltage stability margins that are unseen by the MLP during training.

- Sajan et al. [13] follows a similar approach as in [11] with the difference of using a Genetic Algorithm based Support Vector Machine (SVM) approach for online monitoring of long-term voltage stability. The main reason why machine learning methods are extremely useful in voltage stability assessments is due to the fact that traditional CPF methods are not fast enough for real-time voltage stability evaluation of the system. The authors improved accuracy and minimized the training time of the training cycle based on SVM parameter filtering using genetic algorithm. Voltage magnitude and phase angle obtained from PMU data are used as input data while the voltage stability margin is the estimated output of the SVM algorithm. The effectiveness of the model was initially tested using the New England 39 bus system and further elaborated using the Northern Indian system model, where even for the larger system the model showed to be fast enough for real-time voltage stability assessment.

- Zhang et al. [14] proposes a relatively new and promising learning algorithm called extreme learning machine (ELM) for prediction of voltage stability margin. The stability margin of the system is usually measured in terms of active power changes and the traditional CPF techniques do so by tracing the system P-V curves as load and generation increase. For online monitoring operation, as mentioned in all previous papers, model based methods are slow and cannot be used for more than offline studies. As machine learning becomes more mainstream in all areas of power system application studies, so does the comparison of model-free algorithms. The majority of study cases found in voltage stability assessment using machine learning algorithms are limited to the prediction of voltage stability margins on different operating points under one specific loading direction. With the increase of renewable sources being integrated into the system, loading direction based on forecast starts to become uncertain and thus hinders the estimation capabilities of the latter cases. The authors address the advantages found in their proposal by pointing out the fast learning capabilities of ELM voltage stability margin method when dealing with huge input training data. An ensemble of ELM's for increased robustness and accuracy was also showcased in the work.

- Cai et al. [15] proposed a novel data-based learning and control method for long-term voltage stability. Spatial-temporal data is used as the input of their machine learning model, utilizing principal component analysis as a filter and data size reduction procedure. Artificial neural network is used to build a classifier to reinforce the relationship directly between the system dynamics and optimal control actions.

Simulation results on the 6-bus system, New England 39-bus system and Iceland 189-bus system are given to show the potential of this method for on-line control.

1.4 Motivation

Machine learning tools can be very helpful for the electrical system operator and utility companies, especially for system planning and online monitoring. In this work, five different machine learning algorithms are used to map the voltage and loading levels of the critical bus voltage for changes in the reactive power limits of PV buses of the load increasing system. Principal component analysis (PCA) was used to generate distinguished voltage controlling areas for the IEEE 14 and 118 bus systems in order to finally compare the voltage and loading predictions generated by the Adaptive Network based Fuzzy Inference System (ANFIS), Artificial Neural Networks (ANN), K-Nearest Neighbors (KNN), Support Vector Regression (SVR), and Decision Tree (DT).

The motivation of the present work is to implement a critical bus voltage and load mapping tool that can give supplementary advice to the grid operator. All algorithms, after the training period, are capable of fast prediction calculations due to their trained hypothesis. As expected from machine learning algorithms, once the algorithm is trained, one can use it to quickly generate output predictions for different input values. The reasoning behind the choice of comparing all the aforementioned algorithms for a Critical Voltage and Load predictive tool is that each and every algorithm has its advantages and disadvantages, therefore determining the best overall prediction algorithm for our purpose is key. We wanted to test ANFIS's results against the other four regression methods to see if ANFIS is a viable tool in critical voltage and load mapping.

ANN is the most widely used and known machine learning algorithm, known for its generalization properties, ANFIS is a modified version of the traditional model that handles well uncertainty and imprecision, KNN is simple and intuitive to understand and has no explicit model to build, SVR while not as popular as its classification counterpart has been shown to have good estimation of real-value functions, and DT was proven to be effective even when provided data with missing values [16, 17, 18, 19].

1.5 Thesis Organization

This work is divided in the following chapters:

- Chapter 2 proposes a new method of determining Voltage Area Identification based on PCA. Both IEEE 14 and 118 bus systems are addressed.
- Chapter 3 is a brief explanation of the ANN and ANFIS methods.

- Chapter 4 describes the quasi-static method called Continuation Power Flow which is used to determine the data structure used in the regression algorithms.
- Chapter 5 applies the proposed ideas of chapter 4 to the larger system IEEE 118 bus, as well as comparing the results from all five machine learning algorithms.
- Chapter 6 concludes the work done, in this thesis, and highlights the scope for future research in this area.
- Chapter 7 is the list of published papers during the masters program.

2 Voltage Area Identification

2.1 Introduction

Voltage Stability is a dynamic process dependent strongly on the load profile throughout time. In spite of that, such a problem can be analyzed indirectly with the help of static power flow equations. As a result of a slow dynamical response, voltage stability area identification can be evaluated with the continuous power flow equations [6, 20]. The power system can be separated into coherent buses that exhibit similar responses to reactive power variations. Therefore, the criterion for voltage control area separation is to group similar buses with similar voltage-reactive power responses. After that, the control of the voltage levels can be made through reactive power sources in specific areas. Thus, the compilation of these coherent buses into voltage controlling areas should be simple and non-overlapping [21].

The purpose of this chapter is to explain how we determined the most critical buses with respect to voltage stability. The first section 2.2 will explain the Principal Component Analysis (PCA) method followed by subsection 2.2.1, which is a guided PCA example that in the authors opinion is helpful to illustrate PCA. Later on in section 2.3, PCA will be used to separate the system into coherent voltage stability buses using standard test systems IEEE 14 and IEEE 118 bus, which will be the basis for further chapters along this thesis.

2.2 Principal Component Analysis

Principal Component Analysis (PCA) [14] is one of the oldest and most well-established technique for dimensionality reduction and multivariate analysis. It is widely used in Machine Learning applications, including data compression, image processing, visualization, exploratory data analysis, patterns recognition, and time series prediction. One of the biggest advantages of PCA is that it is a linear transformation of high dimensionality data into a compressed lower dimensionality matrix. Due to its linear nature, compression and decompression are easily achieved via matrix multiplication.

PCA has a very useful property, the first principal component possesses the largest variance among all principal components from the original data matrix, the second principal component has the second largest variance and so forth [22, 23]. Each principal component is an eigenvector $\in \mathbb{R}^{m\times 1}$ that is associated with its respective eigenvalue $\in \mathbb{R}$, where m is the number of features within the information matrix. The eigenvectors are

unit vectors that point to different directions and maximize the variance of the projected information matrix.

To put PCA into perspective, the technique can be thought of as fitting an ellipsoid into a data matrix in which both have the same dimension. Not all of the fitted ellipsoid's axes dimensions are equal, enabling an axis size ranking out of the ellipsoid's dimensions. Each principal component points out to the direction of one of these axes, providing information of the variance along each axis. By omitting the small axes and their respective principal components, only a few percentage of the information is lost with the reduction procedure [24].

Consider a data matrix represented as following

$$M_{data} = \begin{bmatrix} x^{(1)} & x^{(2)} & x^{(3)} & \cdots & x^{(n)} \end{bmatrix} \tag{2.1}$$

$x^{(i)} \in \mathbb{R}^{m\times 1}$ for $i = 1, \cdots, n$ is a column vector in which n is the number of examples and m is the number of features in the M_{data} matrix. The first step of the PCA algorithm is data pre-processing, i.e. feature scaling and mean normalization. Scaling ensures that every feature in the data matrix has comparable range of values and mean normalization centers the data on the origin. The next step is the computation of the covariance matrix

$$\Sigma_{cov} = \frac{1}{m}\sum_{i=1}^{n}(x^{(i)})(x^{(i)})^T \tag{2.2}$$

through singular value decomposition (SVD), in order to retrieve the eigenvalues and the corresponding eigenvectors. Each eigenvector is a unit vector on a different axis, perpendicular to one another for the n-dimensional data matrix. Once the eigenvalues are determined, the ratio between the first kth eigenvalues and the sum of all eigenvalues

$$V_X = \sum_{i=1}^{k} S_{ii} \Big/ \sum_{i=1}^{m} S_{ii} \tag{2.3}$$

quantifies how much of the variance has been retained, i.e, how much of the original information has been saved. In Eq. (2.3), S_{ii} are the diagonal elements of the eigenvalue matrix, k is the number of the chosen principal components and m are the number of features on the data matrix. Once the ratio is determined, the final step in the PCA algorithm is matrix reduction.

The reduced order data matrix is then expressed as

$$Z = U^T M_{data} \tag{2.4}$$

in which $U \in \mathbb{R}^{m\times k}$ is the reduced order eigenvector matrix.

For the purpose of the present work, instead of using PCA to reduce information to its principal components, the elements of the principal components are used for Voltage Area Identification. More details on how to apply PCA and hands-on examples can be found in [25].

2.2.1 Visual Representation of PCA

The following example and figures are used to guide the reader through a visual representation of PCA, which in the authors opinion, is the best way of understanding such topic. As mentioned in the previous subsection, PCA is a data driven hierarchical coordinate system that is used to represent the statistical variations in the data set. Hierarchical in a sense that the algorithm ranks the variance from most to least in this new coordinate system. This means that we are looking for the dominant combination of features from the data set that describe as much of the data as possible. It is visually represented as trying to fit an ellipsoid $\in \mathbb{R}^n$, where n is the dimension of the data's features.

With that being said, let's take a look at the following example. We have a 2D toy data set, where the data points where randomly generated in Matlab through a Gaussian distribution. The reason why we chose to have this type of distribution in the data is to have a compact set of data points for better visual representation. There are 1000 points generated in plot a) all of which centered at the origin $(2, 1)$. The signal is shown in Fig. 1 a) and the resulting principal components are shown in Fig. 1 b) by the vectors $\vec{u_1}$ and $\vec{u_2}$.

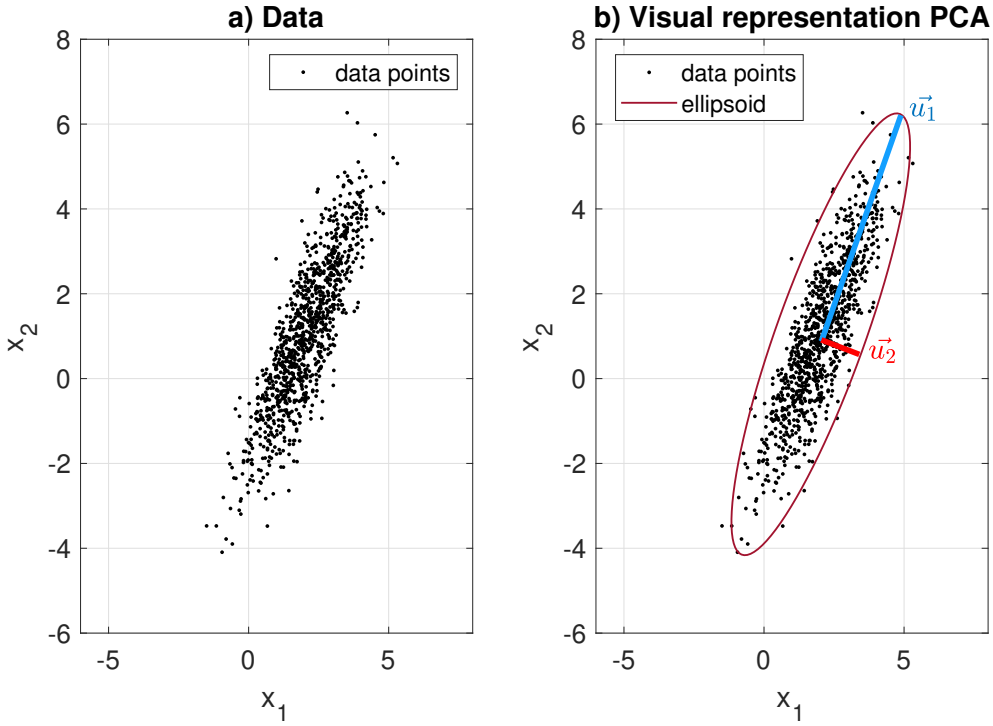


Figura 1 – Visual example where: a) data set, and b) ellipsoid and principal components

The number of principle components reflect the dimension of the data set, which in our case is 2 dimensional. Vectors $\vec{u_1}$, and u_2 and both perpendicular to one another are the principle components of this data set. By visual inspection of Fig. (1) b), it is

clear that the blue vector $\vec{u_1}$ is the one pointing in the direction of the biggest variance and the red vector $\vec{u_2}$ is pointing in the direction of the smallest variance.

2.3 PCA on Power Flow Jacobian

The Newton-Raphson power flow formulation describes the relation between active/reactive power mismatch and angular/voltage variations of the system buses respectively. The power flow equation is described as

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} H & N \\ M & L \end{bmatrix} \cdot \begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix} \tag{2.5}$$

in which $(\Delta P, \Delta Q)$ represents active and reactive power mismatch vectors respectively, $(\Delta\theta, \Delta V)$ represents angular and voltage variation vectors and (H, M, N, L) are partial derivative submatrices that compose the Jacobian Matrix [26]. Since voltage levels are dependent on reactive power variance, mapping the relation between voltage and reactive power can be obtained, once active power variations are disregarded. Therefore, the power flow equation with $\Delta P = 0$ becomes

$$\Delta Q = (L - MH^{-1}N)\Delta V \tag{2.6}$$

in which the difference of the submatrices in Eq. (2.6) is the sensitivity matrix J_{SQV} that expresses the effect of reactive power variation on bus voltage. This means that the inverse of J_{SQV} provides the voltage-reactive power sensitivity information matrix. The J_{SQV}^{-1} matrix can be expressed through the right eigenvectors Ψ, left eigenvectors Φ and the eigenvalues Λ of the J_{SQV} matrix, according to

$$\Delta V = (\Phi\Lambda^{-1}\Psi^T)\Delta Q \tag{2.7}$$

Both eigenvector matrices and the eigenvalue matrix are dimensionally equal and square [20]. The voltage-reactive power sensitivity matrix can be rewritten as

$$\frac{\Delta V}{\Delta Q} \approx \begin{bmatrix} \sum_{i=1}^{p} \frac{\theta_{i1}\psi_{1i}}{\lambda_i} & \sum_{i=1}^{p} \frac{\theta_{i1}\psi_{2i}}{\lambda_i} & \cdots & \sum_{i=1}^{p} \frac{\theta_{i1}\psi_{pi}}{\lambda_i} \\ \sum_{i=1}^{p} \frac{\theta_{i2}\psi_{1i}}{\lambda_i} & \sum_{i=1}^{p} \frac{\theta_{i2}\psi_{2i}}{\lambda_i} & \cdots & \sum_{i=1}^{p} \frac{\theta_{i2}\psi_{pi}}{\lambda_i} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{p} \frac{\theta_{ip}\psi_{1i}}{\lambda_i} & \sum_{i=1}^{p} \frac{\theta_{ip}\psi_{2i}}{\lambda_i} & \cdots & \sum_{i=1}^{p} \frac{\theta_{ip}\psi_{pi}}{\lambda_i} \end{bmatrix} \tag{2.8}$$

in which θ_{ij} is an element of the right eigenvector, where the first subscript relates to the eigenvalue i and the second one relates to the bus number j of the system. Similarly, ψ_{kl} is an element of the left eigenvector, where the first subscript relates to the bus number k of the system and the second one relates to the eigenvalue l.

Each element in the column ξ ($1 \leq \xi \leq p$) of the voltage-reactive power sensitivity matrix, Eq. (2.8), is the voltage sensitivity of a particular bus of the system with respect to the injection of reactive power at bus ξ. Contrarily, each element in the row ζ ($1 \leq \zeta \leq p$) is the voltage sensitivity of the bus ζ with respect to reactive power injection at a particular bus throughout the system. In other words, the rows are the voltage sensitivity of a bus ζ with respect to the reactive power injection in all of the buses and the columns are the voltage sensitivity of all the buses with respect to the reactive power injection at bus ξ [20, 27].

The analysis of the matrix in the RHS of Eq. (2.8) reveals important information for the PCA implementation. Recalling that the columns of J_{SQV}^{-1} are the voltage sensitivity of all buses with respect to reactive power injection into bus ξ, the features of the data matrix are the rows and the different examples for these features rely on the columns of the matrix. Since the data is centered through pre-processing, the principal components multiplied by their respective eigenvalues are vectors that point to maximum information directions in orthogonal coordinates.

2.3.1 IEEE 14 & 118 bus system voltage area division

The voltage division of the IEEE 14 & 118 bus systems are established near the region of voltage instability. The response of the system voltage levels throughout the increase of the load is determined with a continuous power flow analysis. This load is a variant parameter that depicts the dynamics of the system, therefore the insertion of load variation reports the increase of the peak demand. More details on the continuation analysis can be found in review paper [6].

Applying PCA to the Jacobian matrix of the IEEE 14 bus system resulted in an eigenvector matrix and an diagonal eigenvalue matrix, both square matrices with the same dimension. The eigenvalue index indicates that the data matrix can be reduced to a one dimensional model, where the first principal component carries 99.91% of the information of J_{SQV}^{-1}. This means that the first principal component is more than enough information to carry out a voltage area division. As a result of pre-processing, all principal components are centered at the origin and each element of the principal components is a vector that starts at the origin and points to an axis that is related to a bus.

The result of representing each element (bus) in a bar plot is exhibited in Fig. (2). An analysis of Fig. (2) shows a clear separation between positive and negative values of the elements of the principal component. The positive elements are related to the most voltage-reactive power sensitive buses and the negative elements are related to the least ones. This attribute is consistent with the tangent vector ranking utilized to determine

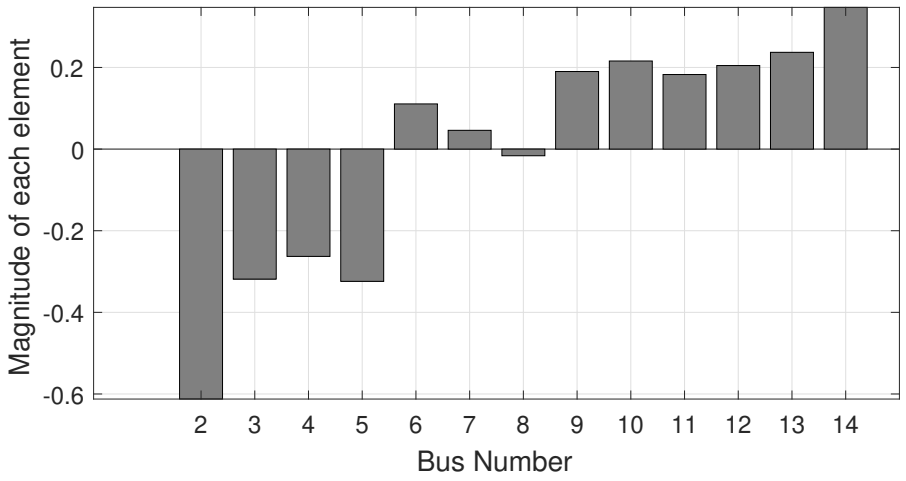


Figura 2 – First Principal Component elements for the IEEE 14 bus system.

the order from most to least critical bus in the system. In this case, bus 14 is the most voltage sensitive bus while bus 2 is the least voltage sensitive bus of the system. Although the IEEE 14 bus system can be separated into two different voltage controlling areas, its compact size allows it to be studied as a unit [18].

The separation of the IEEE 118 bus system into voltage controlling areas with PCA presents its own difficulties, mainly because the first principal component carries out 37.99% of the information from the voltage-reactive power sensitivity matrix. This means that utilizing only the first principal component to divide the system introduces an 62.01% error, which is extremely imprecise for voltage area division. Nevertheless, it is possible to sum the product of the first k principal components with their respective eigenvalues in order to improve the voltage separation property of PCA. Instead of pointing to the most dominant characteristics of J_{SQV}^{-1}, as observed with the IEEE 14 bus system, the vector points to the resulting effect of the k most dominant characteristics of the voltage-reactive power sensitivity matrix. After the summation is done, the resulting vector is normalized. The sum of the first thirteen principal components of the IEEE 118 allows a single vector representation of 90.25% of the information in the data matrix. This is feasible as long as the eigenvectors start at the origin, which is made possible through pre-processing. The distinction in element value and element sign in Fig. (3) identify with precision the similarity in voltage-reactive power buses. Generally, the majority of buses of the IEEE 118 system are attached to nearby numbered buses, hence enabling a separation of the whole system into 6 distinct areas based on element pattern recognition. The ordered bus plot in Fig. (2) and Fig. (3) clearly show that the magnitudes and sign of the modes automatically ranks and separates the buses of the system in per area and in between areas. The change in mode signal from one bus to next one depicts difference

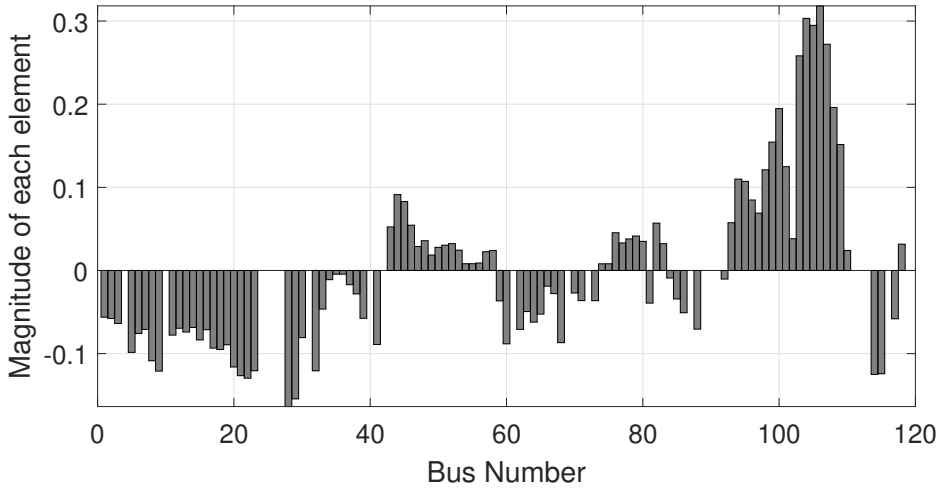


Figura 3 – Resulting vector of Principal Components for the IEEE 118 bus system.

in how voltage reacts with regards to reactive power injection into the system. In other words, there are groups of buses that have their voltages react in a particular way, which can be seen by the 6 distinct areas from the IEEE 118 bus system modal analysis.

Tabela 1 – Voltage Area Division for the IEEE 118 bus system.

Area	Bus Number
1	93 to 112 and 106
2	84 to 92
3	74 to 83 and 118
4	59 to 73, 81 and 116
5	43 to 58
6	1 to 42, 113 to 115 and 117

Throughout all the areas divided by principal components as shown in Tab. 1, area 1 has the largest element relative to bus number 106 which is the critical bus of the system. The consequent largest elements are within area 1 which makes it the most critical area among all. Contrarily, area 6 has the least critical bus among all buses of the system, justified by the smallest element in the resulting principal component. The following smallest elements are within area 6, which makes it the least critical among all areas of the system. The original IEEE 118 bus system reaches its critical voltage point before all the generators reach their reactive power limits. Therefore not all generation buses become PV buses at the critical load, which justifies the absence of some elements in Fig. (3). Although the model lacks a few elements, these elements can be added to the voltage division areas with reasoning, as seen in Table 1.

3 Machine Learning Algorithms

3.1 Introduction

So far we have defined the critical voltage buses in both IEEE 14 and 118 bus systems and their grouping. Bus number 14 is the most critical bus in the IEEE 14 bus system while bus number 106 is the most critical bus in the IEEE 118 bus system. This chapter will lay the foundation of two of the five regression algorithms that are tested in this work, which are ANN and ANFIS. The reason is that the original version of paper [18] was intended to only draw a comparison between ANN and ANFIS. One of the reviewers requested more testing with other methods, namely KNN, SVR and DT. The author does not have proper experience with KNN, SVR, and DT and thus will limit this chapter to only ANN and ANFIS.

It is important to mention that Adaptive Neuro-Fuzzy Inference System (ANFIS) method is the method we want to compare against other regression algorithms in order to check whether or not it is a viable machine learning algorithm for voltage and load mapping. The reason is that ANFIS is not widely used and exposed in the academia, although it has great potential due to fuzziness and the ability to generate rules out of data. The rules, which are verbal expressions, are easily understood and can help system operators with fast and simple solutions.

3.2 Machine Learning general notion

Artificial Intelligence as best described by Andrew Moore, Former-Dean of the School of Computer Science at Carnegie Mellon University, is "The science and engineering of making computers behave in ways that, until recently, we thought required human intelligence." Although industry and marketing interchangeably use the term machine learning and artificial intelligence when it most convenient, there is a big difference in the philosophy behind them. Machine Learning (ML) is a branch of the greater artificial intelligence scope that is focused in learning through experience by automatically finding patterns that explain the relationship between a certain input and output [28]. Figure (4) illustrates the ranking between the broader artificial intelligence area and one of its most famous subareas: Machine Learning.

Machine Learning can be divided into two subareas, *supervised learning* and *unsupervised learning*. Supervised learning is an area of machine learning that is focused in modelling the dependency of an input data set with respect to its output based in labeled

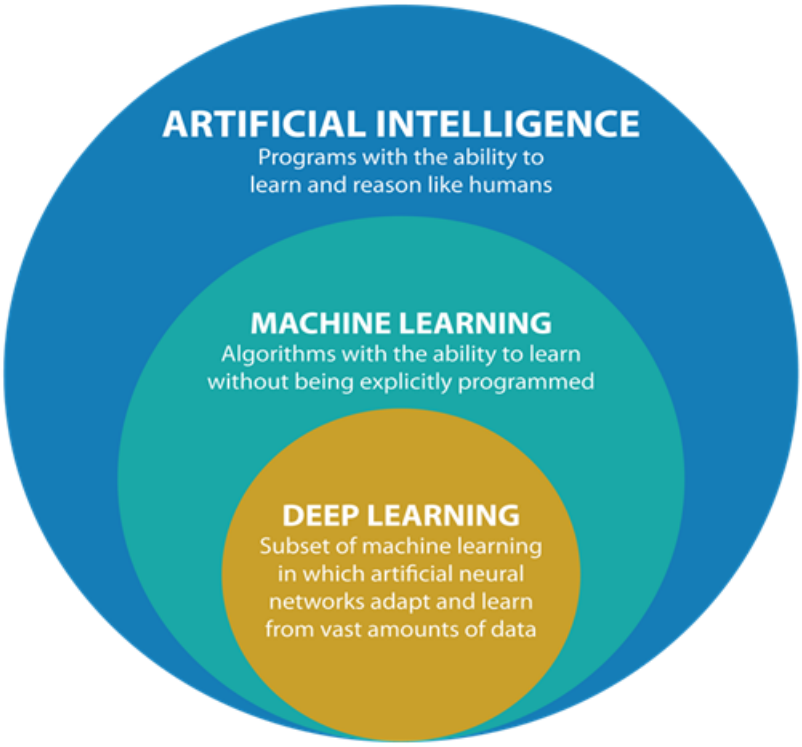


Figura 4 – Artificial Intelligence and its subareas

information. Unsupervised learning, on the other hand, is an area of machine learning that is focused in finding structure in unlabeled data.

This chapter is a birds eye view on two of the five machine learning methods used in this work. The following subsections will briefly describe ANN and ANFIS, in the following order:

- Subsection 3.3 - Artificial Neural Networks (ANN)
- Subsection 3.4 - Adaptive Neuro-Fuzzy Inference System (ANFIS)

3.3 Artificial Neural Networks (ANN)

Artificial Neural Network (ANN) is an artificial intelligence algorithm within the subarea of Machine Learning that, as the name implies, attempts to simulate the functioning of the human brain. This is done by observing the neurons and their interconnections in order to process information [24].

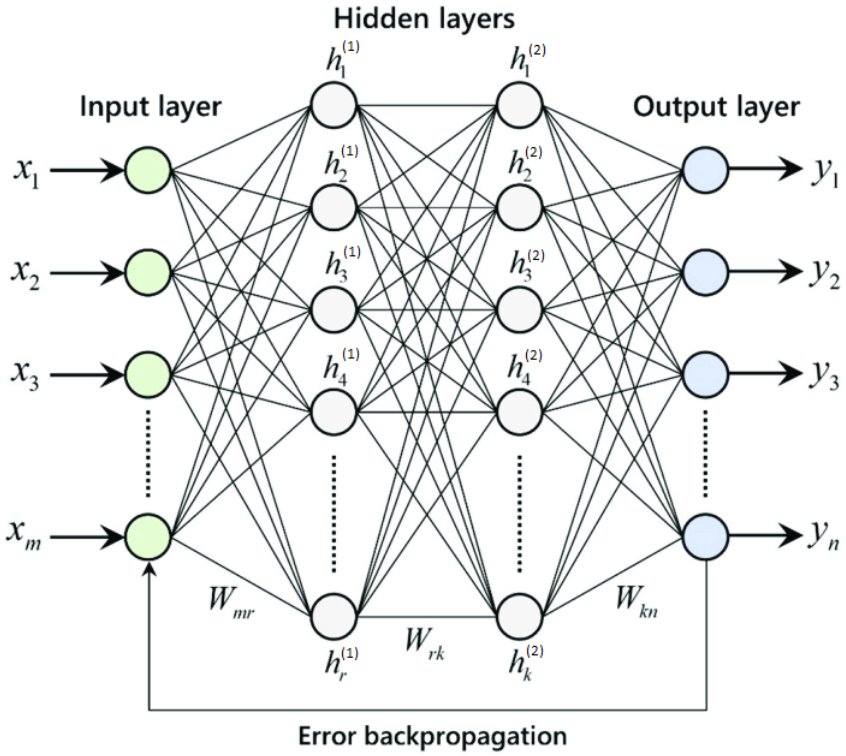


Figura 5 – Artificial Neural Network Architecture

Figure (5) is a visual representation of the architecture of perceptron based neural network, which is one of the machine learning models used in this comparison study. This neural network implementation has three distinct layers: input layer, hidden layer, and output layer. It might seem daunting at first but the neural networks is nothing more than various artificial neurons connected between each other through a pre-defined path. From the figure above, the reader can observe that there is no elements lacking connections and that all circle elements are fully linked to all the elements from the previous layer and the next layer, with the exception of the first and last layers.

The input layer, represented by the green circles in Fig. (5), is the start of the model where we feed the neural network with the features x_i that we chose and its set of data. The hidden layer, represented by the white circles in Fig. (5), is where the computational power comes from and it represents the neuron itself. Each artificial neuron is labeled as h_i^j where i is the number of neurons within a specific layer j of the hidden layer. The number of layers within the hidden layer is greatly dependent on the complexity of the

problem. In our example, there are two layers of artificial neurons inside the hidden layer. The output layer is also a layer of neurons limited to no further connections and used to output the final weighted signal.

The following subsection describes the classical structure of an ANN and was extracted from the reference book [29].

3.3.1 Classical ANN

The basic structure of an ANN is the artificial neuron shown in Fig. (6), which resembles the biological neuron in its shape and function [30].

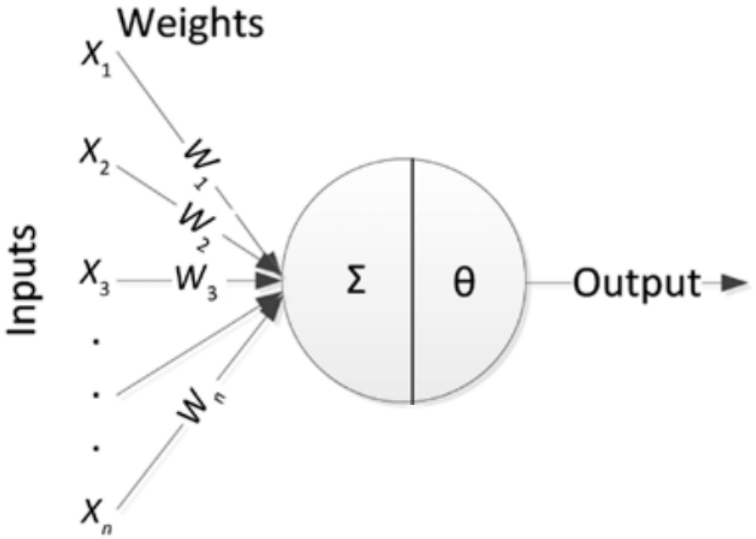


Figura 6 – Artificial Neuron

The inputs (X) are connected to the neuron through weighted connections emulating the dendrite's structure, whereas the summation, the bias (b), and the activation function (θ) play the role of the cell body, and the propagation of the output is analogous to the axon in a biological neuron. Mathematically, a neuron is equivalent to the function:

$$Y = \theta\left(\sum_{i=1}^{n} W_i X_i + b\right) \tag{3.1}$$

which can be conveniently modeled, using a matrix form,

$$Y = \theta(W.X + b) \tag{3.2}$$

where $W = \begin{bmatrix} W_1 & W_2 & \cdots & W_n \end{bmatrix}$, and $X = \begin{bmatrix} X_1 & X_2 & \cdots & X_n \end{bmatrix}^T$.

The activation function shapes the output or state of the neuron. There are multiple activation functions in the literature, but we chose to use the sigmoidal activation function, also known as logistic activation function. It has the following formulation:

$$\text{Sigmoid function: } \theta(a) = \frac{1}{1 + e^{-a}} \tag{3.3}$$

A neural network is simply an association of cascaded layers of neurons, each with its own weight matrix, bias vector, and output vector. If an input vector is constituted of N inputs and a layer of M neurons, W_{ij} represents the weight of the connection of the j th input to the i th neuron of the layer; Y_i and b_i are, respectively, the output of and the bias associated with the j th neuron. A layer of neurons can be conveniently represented, using matrix notation, as follows:

$$W = \begin{bmatrix} W_{11} & \dots & W_{1M} \\ \vdots & \vdots & \vdots \\ W_{N1} & \dots & W_{NM} \end{bmatrix} \tag{3.4}$$

The row index in each element of this matrix represents the destination neuron of the corresponding connection, whereas the column index refers to the input source of the connection. Designating by Y the output of the layer, you can write

$$Y = \begin{bmatrix} Y_1 \\ \vdots \\ Y_i \\ \vdots \\ Y_N \end{bmatrix} = \begin{bmatrix} \theta\left(\sum_{j=1}^{M} W_{1j}X_j + b_1\right) \\ \vdots \\ \theta\left(\sum_{j=1}^{M} W_{ij}X_j + b_i\right) \\ \vdots \\ \theta\left(\sum_{j=1}^{M} W_{Nj}X_j + b_N\right) \end{bmatrix} = \theta(W \cdot X + b) \tag{3.5}$$

To aid in identifying the layer corresponding to a particular matrix, superscript indexes are used. Thus, W_n^k represents the weight of the connection between the j th neuron in layer k -land the i th neuron in layer k, and Y_i^k is the output of the i th neuron of the k th layer. The network output is the output of the last layer (also called the output layer), and the other layers are called hidden layers. For a network with two hidden layers, as was illustrated in the previous section in Fig. (5), results in a output function

$$\mathbf{Y_{out}} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} = \theta\left(W^3\theta\left(W^2\left(\theta\left(W^1X + b^1\right)\right) + b^2\right) + b^3\right) \tag{3.6}$$

The training step requires constant update of the weights with gradient descent during back propagation, and the output $\mathbf{Y_{out}}$ should match the target T_i and minimize the mean squared error

$$E = \frac{1}{2}\sum_{i=1}^{p}\left\|Y_i - T_i^2\right\| \tag{3.7}$$

where Y_i is the output obtained by propagating input X through the network.

3.4 Adaptive Neural Fuzzy Inference System (ANFIS)

ANFIS is the merge of Artificial Neural Networks (ANN) and Fuzzy Inference System (FIS). In real world problems, a great deal of information and data are obtained from the human expert. This set of information allows ANFIS to be used for modelling, controlling and parameter estimation.

The benefit of using an ANFIS based model in comparison to a fuzzy logic model is to train the algorithm without relying solely on expert knowledge, and, in comparison to an ANN, the model does not rely solely in numerical data. In addition ANFIS based model compared to other generalization models are: adaptation capability, ability to generate nonlinear hypothesis and rapid learning capacity [31]. Thus a rule-based fuzzy logic model is achieved and the rules are developed during the training process. Based in our data ANFIS constructs a FIS in which the membership function parameters are obtained through iterations of the algorithm.

The two most commonly used FIS are Mamdani and Sugeno [32], their fundamental difference is that Mamdani requires output membership functions while Sugeno generates linear or constant outputs relying only on the input membership functions. In the present analysis, the Sugeno-type fuzzy inference system is used because the provided data comes from numerical simulation of an automatic rule generator [33]. For the description of ANFIS architecture, it was assumed two inputs: x_1 and x_2. Accordingly two fuzzy if-then rules for the 1st-order Sugeno Fuzzy Model are expressed as:

Rule 1: If x_1 is A_1 and x_2 is B_1, then $f_1 = p_1x_1 + q_1x_2 + r_1$,

Rule 2: If x_1 is A_2 and x_2 is B_2, then $f_2 = p_2x + q_2y + r_2$,

in which A_i and B_i are the fuzzy sets, f_i is the output, and p_i, q_i, and r_i are the design parameters that are determined during the training process. The details of the method below can be found in review paper [34, 35, 36].

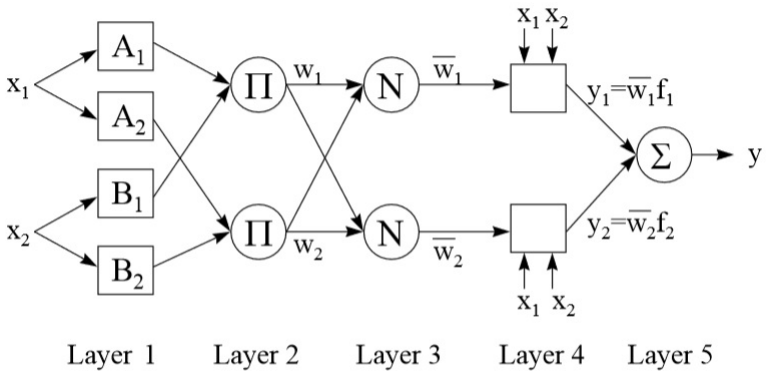


Figura 7 – ANFIS arquitecture two-input, two-output first order Sugeno Model.

The ANFIS architecture with two rules is shown schematically in Fig. (7), and consists of five layers. The ANFIS algorithm consists of five layers. Layers 2, 3 and 5 are operation layers meaning that there are no trainable parameters. Layers 1 and 4 consist of two trainable parameter sets, the antecedent membership function parameters and the polynomial parameters also denominated consequent parameters.

The structure of ANFIS is described in the following. First of all in order to compact the formulation, we will define O_i^j as the output of ith node and jth layer thus, Layer 1: Membership Functions in which A_i, B_i are linguistic labels characterized by respective membership functions related to the input:

$$O_i^1 \equiv \mu_{A_i}(x), \quad \textit{for i=1,2} \tag{3.8}$$

Layer 2: Calculation of firing strengths by product:

$$O_i^2 \equiv w_i = \mu_{Ai}(x_1)\mu_{B_i}(x_2), \quad \textit{for i=1,2} \tag{3.9}$$

Layer 3: Firing strengths from previous layer are normalized to determine the strength of each rule from the total firing strength:

$$O_i^3 \equiv \overline{w_i} = \frac{w_i}{w_1 + w_2}, \quad \textit{for i=1,2} \tag{3.10}$$

Layer 4: In the ith node the contribution of the ith rule to the overall output is calculated:

$$O_i^4 \equiv \overline{w_i} f_i = \overline{w_i}(p_i x_1 + q_i x_2 + r_i) \tag{3.11}$$

in which $\{p_i, q_i, r_i\}$ is the parameter sets mentioned above and $\overline{w_i}$ are the normalized firing strengths from Eq. (3.10).

Layer 5: The total contribution of all inputs from layer 4 is:

$$O_i^5 \equiv \sum_i^n \overline{w_i} f_i = \frac{\sum_i^n w_i f_i}{\sum_i^n w_i} \tag{3.12}$$

Training ANFIS requires the use of a gradient descent algorithm to optimize the antecedent parameters and the least squares algorithm to determine the consequent parameters. The use of two different algorithms inside the structure of ANFIS justifies it to be classified as a hybrid algorithm. The first updated parameters are the consequent $\{p_i, q_i, r_i\}$ and the antecedent parameters are then updated using back-propagation, just as in Artificial Neural Networks [37].

4 Critical Voltage and Load Mapping

4.1 Introduction

So far we have defined the critical voltage buses in both IEEE 14 and 118 bus systems and their grouping. Bus number 14 is the most critical bus in the IEEE 14 bus system while bus number 106 is the most critical bus in the IEEE 118 bus system in chapter 2. In chapter 3, we gave a brief overview on ANN and ANFIS, two of the five machine learning methods used in this work. This chapter will lay the foundation of the continuation power flow in section 4.2, which is an extension of the Newton power flow solver, that continuously traces the voltage level of all buses as loading increases. The continuation power flow solver is used to determine the Voltage-Load Hatched Area, which is the boundary from where we will extract the data used in the regression algorithms. The proposed data structure that is defined in section 4.3 was first implemented for the IEEE 14 bus system and used in subsequent chapters for the bigger IEEE 118 bus system. Section 4.4 is devoted to determining the data structure that will be used throughout the rest of this work. It was initially tested with ANFIS and further explored in chapter 5 with the five regression methods.

4.2 Continuation Power Flow

The traditional Power Flow algorithm is a Newton-Raphson approach to determine the roots of a nonlinear system. For the case of electrical (nonlinear) systems, the solution provides the state variables: electrical bus angle and electrical bus voltage [26].

The contribution of all active and reactive power components for each of the ith buses is described according to,

$$\begin{aligned} P_{Gi} - P_{Li} - P_{Ti} &= 0 \\ Q_{Gi} - Q_{Li} - Q_{Ti} &= 0 \end{aligned} \tag{4.1}$$

in which subscript G, L and T denote bus generation, load and injection, respectively. Also, the contribution of all active and reactive power injections from the ith bus to the jth bus is

$$\begin{aligned} P_{Ti} &= \sum_{j=1}^{n} V_i V_j y_{ij} \cos\left(\delta_i - \delta_j - \vartheta_{ij}\right) \\ Q_{Ti} &= \sum_{j=1}^{n} V_i V_j y_{ij} \sin\left(\delta_i - \delta_j - \vartheta_{ij}\right) \end{aligned} \tag{4.2}$$

in which V_i and V_j are the voltages of bus i and j, δ_i and δ_j are the bus angles of bus i and j, respectively, and y_{ij} is the $(i,j)^{th}$ element of admittance matrix Y_{BUS}.

On the other hand the continuation algorithm is a path-following methodology used to solve systems of nonlinear equations. For power flow, the purpose of the continuation method is to achieve continuum power flow solutions for a distinct load change scenario. The continuation technique in Power Flow problems requires a variant parameter that depicts the dynamics of the electrical system. Therefore a load parameter is inserted into the equation to report the increase of the peak load demand. Throughout this section we will reproduce shortly the fundamental aspects of the insertion of the load parameter into the Power Flow equations [38].

The load parameter is represented by λ and is defined in the range $0 \leq \lambda \leq \lambda_c$ and the limits correspond to base load problem and critical load, respectively. As $\lambda \to \lambda_c$, the root finding procedure goes close to a saddle-node point, and, except for a proper trajectory, the solution is not determined.

The insertion of the load parameter for each bus of the system into the active and reactive power equations results in:

$$\begin{aligned} P_{Li} &= P_{Li0} + \lambda(k_{Li} S_{\Delta base} \cos\psi_i) \\ Q_{Li} &= Q_{Li0} + \lambda(k_{Li} S_{\Delta base} \sin\psi_i) \end{aligned} \tag{4.3}$$

where P_{Li0} and Q_{Li0} are the original load at the ith bus for active and reactive power respectively, k_{Li} is the rate of load change at the ith bus, ψ_i is the power factor angle of load change at the ith bus and $S_{\Delta base}$ is the apparent power chosen to provide appropriate scaling of λ.

Generation as well can be portrayed with the inclusion of the load parameter, thus being represented as such

$$P_{Gi} = P_{Gi0}(1 + \lambda k_{Gi}) \tag{4.4}$$

in which P_{Gi0} is the base case active generation in bus i and k_{Gi} is the rate of change in generation as λ varies. Substituting Eqs. (4.3) and (4.4) into the power flow Eqs. (4.1) results in

$$\begin{aligned} P_{Gi0}(1 + \lambda k_{Gi}) - P_{Li0} - \lambda(k_{Li} S_{\Delta base} \cos\psi_i) - P_{Ti} = 0 \\ Q_{Gi0} - Q_{Li0} - \lambda(k_{Li} S_{\Delta base} \sin\psi_i) - Q_{Ti} = 0 \end{aligned} \tag{4.5}$$

Once the load parameter is accounted for in the power flow equations as identified in Eq. (4.5), the next step is to apply the continuation algorithm to the reformulated system.

The continuation power flow equations are now expressed as such

$$F(\delta, V, \lambda) = 0, \qquad 0 \leq \lambda \leq \lambda_c \tag{4.6}$$

Solving the new power flow equations requires the continuation algorithm to start from a known solution with the use of a predictor-corrector scheme to find the following solutions at different load levels. The corrector algorithm is a slightly modified Newton-Raphson power flow while the predictor algorithm is an appropriately sized step in the tangent direction of the solution path [39]. The algorithm is stopped when critical point is reached, which is the point of maximum loading value. The tangent component of λ is zero at the critical point and the determinant of the Jacobian matrix is null.

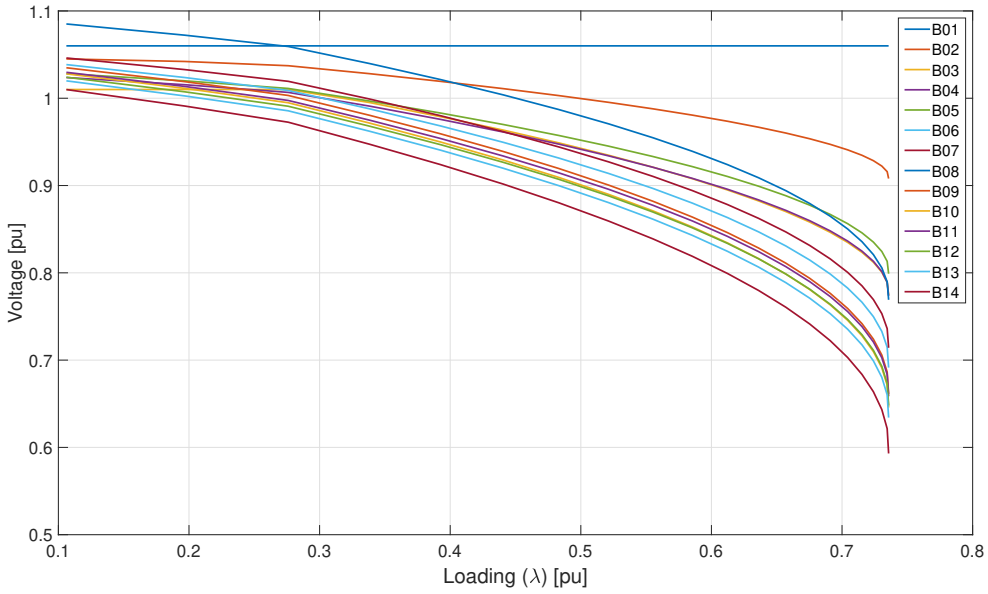


Figura 8 – P-V plot for the IEEE 14 bus system

Figure (8) is the result of running the simulation for the IEEE 14 bus system, known as the P-V curve. As loading increases voltage declines until reaching the voltage critical point at $\lambda = \lambda_c$. For the sake of cleanliness only the CPF plot of the IEEE 14 bus system is displayed in this section.

4.3 IEEE 14 bus Critical Voltage Mapping

We will start by figuring out the best data scheme for voltage and load mapping for ANFIS. As mentioned previously, ANFIS is the machine learning algorithm that we want to test others against to check the its viability in voltage and load mapping assessment tool. That being said, in order to map the voltage in the critical bus using the five regression methods chosen for this study, a minimum number of data is required to validate the results,i.e., to achieve small errors by generalization. The analysis begins with two standard cases: a) the base system and b) the system with double maximum limits of the reactive power Q_{max} (MVar) of all synchronous generators and compensators. The second case will be treated as the modified case.

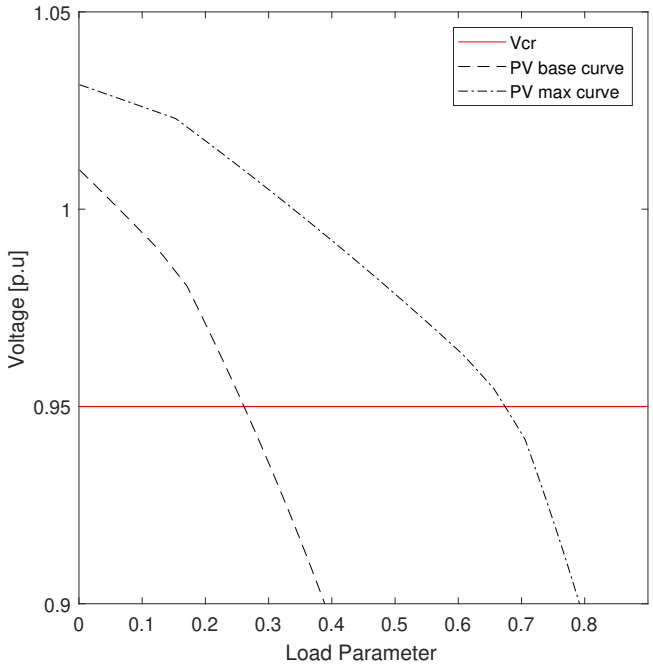


Figura 9 – Critical Bus #14 PV curve for base and maximum case.

Figure (16) exhibits the result from simulation with those two standard cases. Both curves cross the limiting voltage at different load parameter values, the base case crosses the limit voltage at $\lambda = 0.261$ and for the modified case the limit is crossed at $\lambda = 0.674$. The lower limit for the critical bus voltage is 0.95 p.u which is the minimum value admissible for voltage though out the system.

The increase in Q_{max} results in a larger reactive power range of the PV buses which justifies the difference in voltage levels for even loading values between the base case and the modified case.

All other combinations of Q_{max} that are not maximum as mentioned above will result in P-V curves that are in between the base case and the maximum case. Therefore it is possible to map all the voltages and load parameters inside the hatched area between curves depicted in Fig. (10).

If we desire to control the voltage in the critical bus by means of AVR, it is necessary to obtain the model of the voltage output through the whole variation of Q_{max} up to the maximum value stipulated in the study.

4.4 Data for ANFIS

IEEE 14-bus is the chosen system for these tests. It has 4 PV buses in which the maximum limit of the reactive powers Q_{max} are changed for every training example. Then,

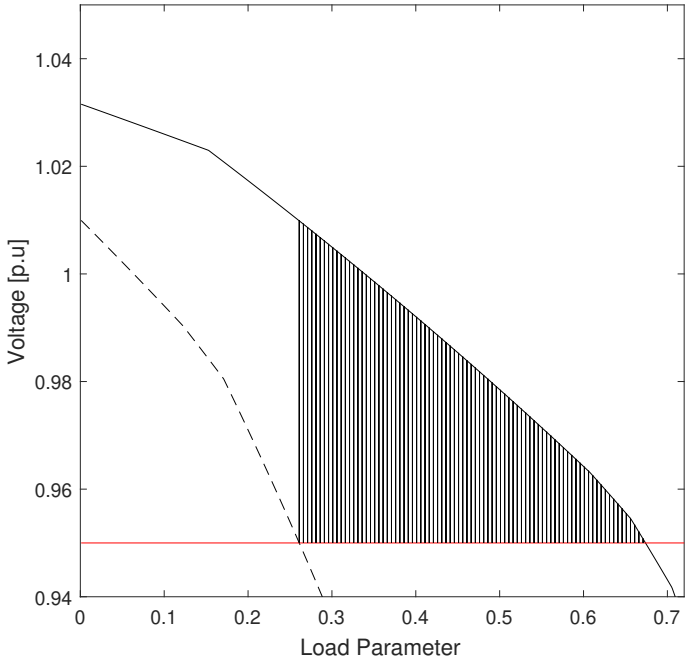


Figura 10 – Hatched area between PV max curve and voltage limit.

for each composition of Q_{max} for the PV buses, input data is sorted in a matrix form such as

$$X_{training} = \begin{bmatrix} Q_3^1 & Q_2^1 & Q_8^1 & Q_6^1 & \lambda^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ Q_3^j & Q_2^j & Q_8^j & Q_6^j & \lambda^j \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ Q_3^m & Q_2^m & Q_8^m & Q_6^m & \lambda^m \end{bmatrix} \tag{4.7}$$

Equation (5.1) is the input matrix used to train the ANFIS algorithm. For each example of the training matrix there are 5 inputs $(Q_3^j, Q_2^j, Q_8^j, Q_6^j, \lambda)$ in which Q_i^j is the maximum reactive power injection from PV buses for ith bus and jth training example and λ^j is the power load for the jth training example. For the sake of compactness, the cross validation input $X_{crossval}$ is not presented, but it has the same configuration as the training input.

The output vector $Y_{training}$ shown in Eq. (5.2) is the voltage tied up to the training input for each jth example.

$$Y_{training} = \begin{bmatrix} U_c^1 & \cdots & U_c^j & \cdots & U_c^m \end{bmatrix}^T \tag{4.8}$$

The superscript j in the elements of $X_{training}$ and $Y_{training}$ denotes the jth row of examples from the training set for a set of m training examples. For the sake of compactness, the cross validation output $Y_{crossval}$ is not presented, but it has the same configuration as the training input, whose kth element is defined as V_c^k.

Once the training set is obtained, the next step in the algorithm is to train the network to minimize the error between hypothesis and actual output. This procedure determines the antecedent and consequent parameters. The data used for each test is chosen in order to compare the hypotheses by varying the number of examples and the training data set.

In the present analysis the comparison index J is defined as the average least square cost function

$$J = \frac{1}{2n}\sum_{i=1}^{n}(h^i - V_c^i)^2 \tag{4.9}$$

in which h^i is the hypothesis generated by the ANFIS algorithm for a specific cross validation set and V_c^i is the output of the cross validation set. The superscript i is the number of the cross validation example with a total of n examples.

Notice that to show the evolution of the results of the tests, seen bellow, the vertical axis of the figures will be presented in different scales.

4.4.1 First Data Test

For the first test, we increase $(Q_3^j, Q_2^j, Q_8^j, Q_6^j) = (50, 40, 24, 24)$ (IEEE 14-bus) individually in increments of 10% up to doubling their values. From all information achieved through the continuation power flow, the useful data is concentrated inside of what we define as the Voltage-Load Hatched Area (VLHA) as shown in Fig. (10). A specific set of data is collected to serve as training data. The main idea of this analysis is to obtain the data at the vertices of the polygon formed by the VLHA and, from those data, predict critical voltage values for any λ contained within the hatched area, i.e. in a reduced range for λ. This procedure accelerates the learning process, consequently the solution is found with less examples.

Figure (11) shows schematically the idea of choosing data from the vertices of the VLHA by varying Q_{max} at bus 6. This chosen training set consists of 79 training examples including the base case. After training with $X_{training}$, a cross validation set with 16 random examples was used to determine the effectiveness of the mapping algorithm. This cross validation set will be used for all tests in order to determine the best $(X_{training}, Y_{training})$.

Figure (12) shows the difference between the output of the cross validation data and the hypothesis generated by the ANFIS algorithm. The error between the hypothesis and the output is large and can be justified by either lack of data examples, model

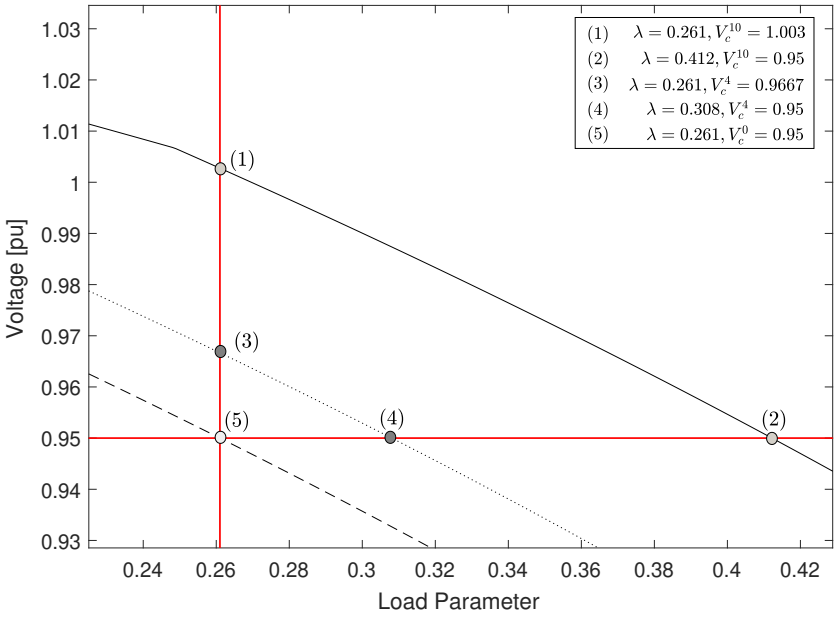


Figura 11 – Example of data collected for base case, 40% increase and 100% increase of reactive power in Q_6.

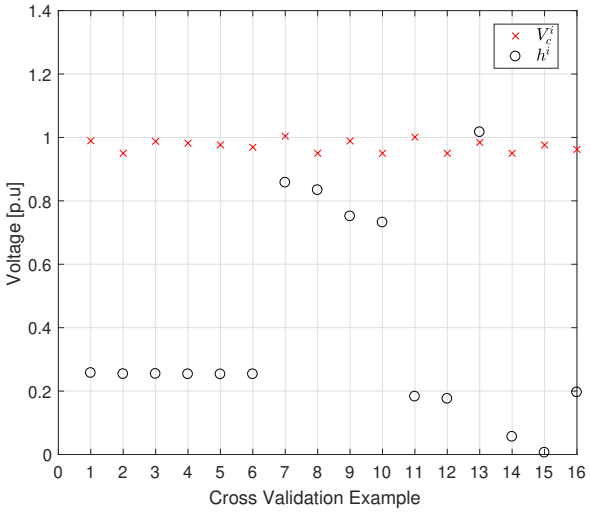


Figura 12 – Comparison between h^i and V_c^i for test 1.

oversimplification or poor choice of training data set. The cost function for the first data set is $J = 0.2181$.

4.4.2 Second Data Test

For the second test, we repeat the same methodology used in the first test, but now with new training examples. The second data set consists of 21 examples where $(Q_3^j, Q_2^j, Q_8^j, Q_6^j)$ gain simultaneous 10% increments until doubling their original values. These 21 training examples are obtained from the vertices of the polygons formed by the

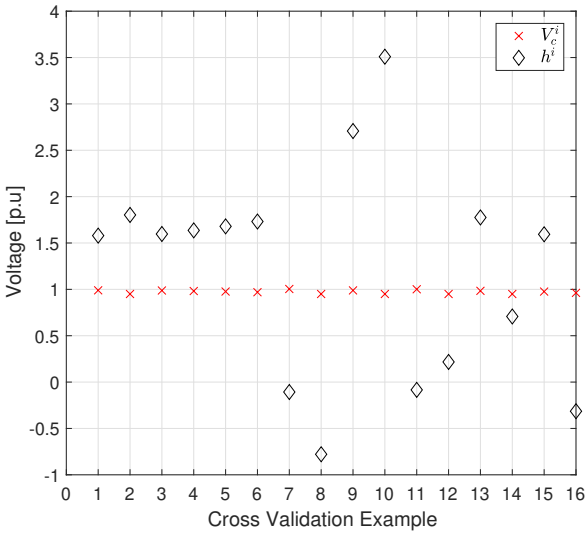


Figura 13 – Comparison between h^i and V_c^i for test 2.

second data VLHA, such as in the previous test shown by Fig. (11).

After training with $X_{training}$, a cross validation set with 16 random examples was used to determine the effectiveness of the mapping algorithm. Figure (13) shows the difference between the output of the cross validation data and the hypothesis generated by the ANFIS algorithm. The error between the hypothesis h^i and the output V_c^i invalidates the use of the second data set as a feasible set of data. The cost function for the second data set is $J = 0.3930$.

4.4.3 Third Data Test

Again, for the third test, the procedure presented in the first data test is repeated with new training data set. For this test, the chosen training set consists of 99 examples: a combination of the first data set with the second data set. The idea is to combine both training sets and analyze whether more data, by its own, would give better results.

We observed that the increase of the number of examples in the data matrix shows to be inefficient when generating a valid hypothesis, unless the proper choice for the training example is adopted. The combined training set resulted in even greater error between the hypothesis h^i and output V_c^i with a cost function value of $J = 7.2239$.

Figure (14) shows the difference between the output of the cross validation data and the hypothesis generated by the ANFIS algorithm.

4.4.4 Fourth Data Test

For the fourth test each example is the combination between $(Q_3^j, Q_2^j, Q_8^j, Q_6^j)$ with gains of 100%, i.e., C_1^4, C_2^4, C_3^4 and C_4^4 for all Q_{max}. The same methodology used in

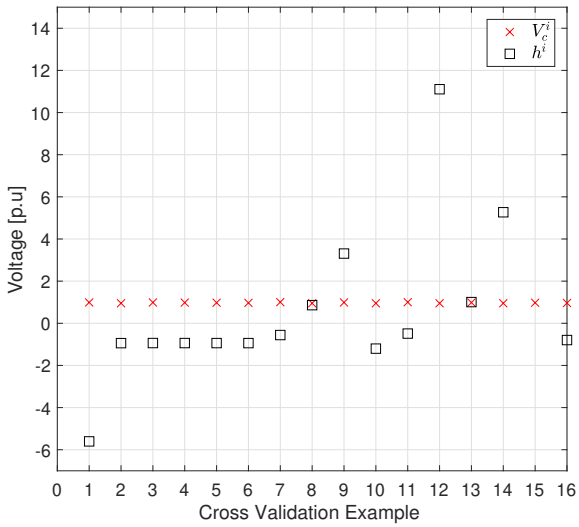


Figura 14 – Comparison between h^i and V_c^i for test 3.

previous tests is used for the fourth test, that means that the chosen training set in the new conditions consists of 31 training examples and the cross validation set is the same as in previous tests.

Figure (15) shows the difference between the output of the cross validation data and the hypothesis generated by the ANFIS algorithm. We observed that, by choosing data that is the combination of Q_{max}, the error between the hypothesis h^i and the output V_c^i is the smallest among all tests with a cost function value of $J = 7.657 \times 10^{-4}$.

This test suggests that with a combination between $(Q_3^j, Q_2^j, Q_8^j, Q_6^j)$ and selecting appropriate data we can estimate the critical bus voltage, with a acceptable error, for any load parameter bounded by VLHA. The reliability of the hypothesis generated by the ANFIS algorithm is attached to the area of the VLHA, implying that for smaller gain steps the algorithm will predict the critical bus voltage with greater precision.

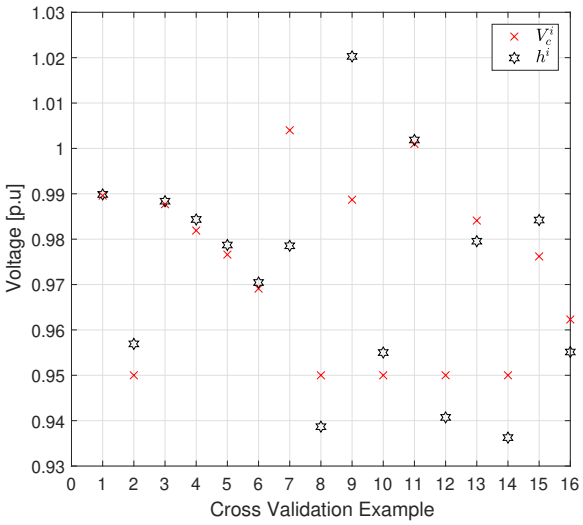


Figura 15 – Comparison between h^i and V_c^i for test 4.

4.5 Data selection conclusion

ANFIS for data driven models with a small data set proved to be effective in predicting the critical bus voltage for the IEEE 14 bus system within the VLHA load parameter range. The small number of examples within the training data set would lead to inaccurate results if trained with a traditional ANN. However, it was overcomed with the addition of fuzziness into the algorithm represented by the input membership functions.

Nevertheless, a proper choice of training data is required in order to model the critical bus voltage. Now that we have decided on the data structure that best suits ANFIS for the IEEE 14 bus system, we will extend this idea to the IEEE 118 bus system and compare it against ANN, KNN, SVR, and DT.

5 Voltage and Load prediction comparison for the IEEE 118 bus

5.1 Introduction

Chapters 2 to 4 have discussed the method for dividing the system into coherent voltage-reactive power buses, briefly mentioned how ANN and ANFIS works and defined a data structure that was tested for the IEEE 14 bus system and will be used for voltage load mapping of a bigger system (IEEE 118) in this chapter.

This chapter contains the voltage-load hatched area for the critical bus of the IEEE 118 bus system as well as the result of the comparison tests and a section dedicated to reviewing and scoring the overall best regression algorithm for the task.

5.2 Voltage-Load Hatched Area for the IEEE 118 system

Critical voltage and load mapping begin with two standard cases: a) the base case and b) the system with an augmented maximum limit of the reactive power Q_{max} (MVar) of the synchronous generators and compensators. As mentioned previously, area 1 was pointed out as the voltage stability critical area, shown by the elements of the resulting principal component in Fig. (3). The increase in the maximum reactive power limits of the PV buses yields a greater critical load profile, elevating voltage levels of nearby buses. Due to the control of the critical bus voltage to be a rather localized problem, reactive power injection efforts are limited by the area in which the critical bus stands. As a result, only the PV buses presented in area 1 exhibit significant impact on voltage and critical load levels of the critical bus. Area 1 of the IEEE 118 bus system has 9 PV buses, of which 4 of them have meaningful impact on the voltage level of the critical bus 106. Following the procedure mentioned above, the augmented maximum limit system chosen for this analysis represents an increase of 25% on all the maximum limits of reactive power in the 4 PV buses ($Q_{103}, Q_{104}, Q_{105}, Q_{107}$). The augmented limit system widens the level of operable load as well as the critical bus voltage, enabling a representation of the generators and compensators for the overly stressed system.

Fig. (16) exhibits the result from simulation with those two standard cases, in which the dot-dash line and the dashed line is the PV curve for the augmented maximum reactive power limit system and the PV curve for the base case respectively. Each curve intersects the limiting voltage at different loading points, $\lambda_{base} = 0.3087$ while $\lambda_{max} = 0.3615$. The red solid line is the minimum value admissible for voltage through

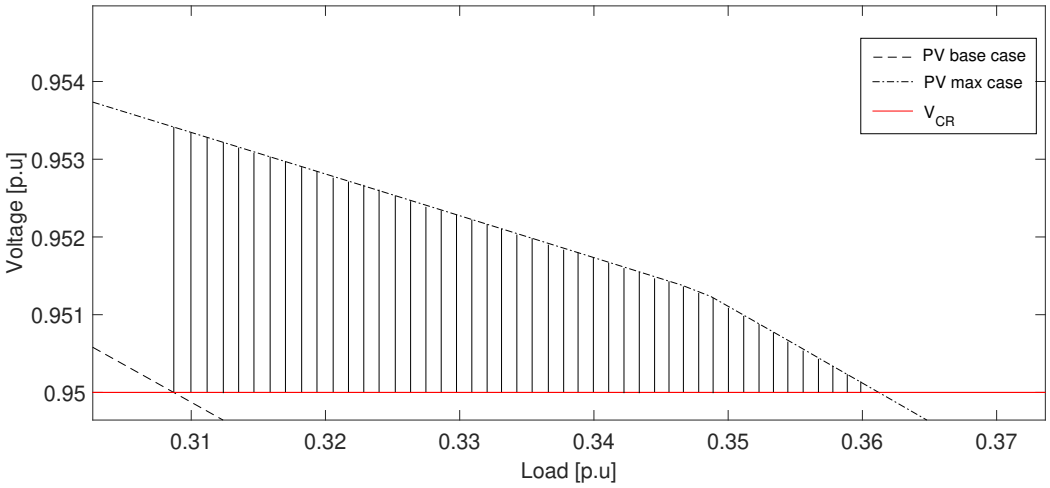


Figura 16 – Critical Bus #106 PV curve for base and maximum case.

out the system, given by 0.95 p.u.

The increase in Q_{max} results in a larger reactive power range of the PV buses which justifies the difference in voltage levels for even loading values between the base case and the modified case. All other combinations of Q_{max} that are not maximum as mentioned above will result in PV curves that are in between the base case and the maximum case. Therefore it is possible to map all the voltages and load parameters bounded by both PV curves and the limiting system voltage, depicted by the hatched area in Fig. (16), defined as Voltage-Load Hatched Area (VLHA) [18].

5.3 Data structure for ANN, ANFIS, KNN, SVR & DT

In order to compare the voltage and load prediction capabilities of all the regression algorithms aforementioned, the first step in the methodology is to determine the data set. A different voltage prediction methodology for the critical bus of the IEEE 14 bus system was presented in [18], it is based on data that was acquired from the vertices of the voltage-load polygon. The proposition is suited for voltage and load predictions because of the near perfect linear voltage response for the increasing load. Once the extremities of the curves that form the polygon are known, all other data points within the polygon can be determined with a linear hypothesis. This polygon is the result of the intersection of two PV curves with the limiting voltage value of the system.

As mentioned previously, the critical area of the IEEE 118 bus system has nine PV buses from which four of them have significant impact on the voltage level of the critical bus 106. Therefore, the training data matrix is the combination between $(Q^j_{103}, Q^j_{104}, Q^j_{105}, Q^j_{107})$

with gains of 25%, i.e., the following combinations C_1^4, C_2^4, C_3^4 and C_4^4 for all Q_{max}, resulting in an $N = 31$ example data matrix. For critical voltage output V_{106}, the input data is sorted in a matrix form such as

$$X_{V_{train}} = \begin{bmatrix} Q_{103}^1 & Q_{104}^1 & Q_{105}^1 & Q_{107}^1 & \lambda^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ Q_{103}^j & Q_{104}^j & Q_{105}^j & Q_{107}^j & \lambda^j \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ Q_{103}^N & Q_{104}^N & Q_{105}^N & Q_{107}^N & \lambda^N \end{bmatrix} \tag{5.1}$$

Equation (5.1) is the input matrix used to train ANFIS, ANN, KNN, SVR, and DT for critical voltage output. For each example of the training matrix there are 5 input features $(Q_{103}^j, Q_{104}^j, Q_{105}^j, Q_{107}^j, \lambda^j)$ in which Q_i^j is the maximum reactive power injection from PV buses for ith bus and jth training example and λ^j is the power load for the jth training example.

As shown below, the output vector $Y_{V_{train}}$ is the voltage tied up to the training input $X_{V_{train}}$ for each jth example.

$$Y_{V_{train}} = \begin{bmatrix} V_{106}^1 & \cdots & V_{106}^j & \cdots & V_{106}^N \end{bmatrix}^T \tag{5.2}$$

Contrarily, for load output λ, the input data matrix $X_{L_{train}}$ is an $N = 31$ example data matrix with $(Q_{103}^j, Q_{104}^j, Q_{105}^j, Q_{107}^j, V_{106}^j)$ input features and an output $Y_{L_{train}}$ matrix that refers to the different loads for each example. For the sake of compactness, $(X_{L_{train}}, Y_{L_{train}})$, the testing inputs for critical voltage and load $(X_{V_{test}}, X_{L_{test}})$ are not presented explicitly, but they exhibit the same configuration as seen in Eqs. (5.1) and (5.2).

Once the training set is established, the next step in the algorithm is to train the network to minimize the error between hypothesis and actual output.

Notice that to show the evolution of the results of the tests, seen bellow, the vertical axis of the figures will be presented in different scales. More details on the regression algorithms with hands-on examples can be found in [16, 17, 35].

5.4 ANN, ANFIS, KNN, SVR & DT voltage and load prediction comparison

In order to thoroughly assess the mapping capabilities of the aforementioned regression algorithms, eight different tests were conducted. All of them were initially evaluated with a 10 fold cross validation algorithm checker used to determine the best input data configuration to hypothesis error ratio. The chosen architecture for the ANN is a two layer feedforward neural network, in which the hidden layer consists of 3 neurons with a single output in the output layer. The reason for a single hidden layer is that one can approximately represent a non-linear function with only one hidden layer, as seen in [40]. This one hidden layer model is known as an universal approximator. The number of neurons was determined by a widely used rule of thumb, which is 2/3 the size of the input layer. It was adopted a tangent sigmoidal transfer function for the hidden layer and a rectified linear unit (ReLU) transfer function for the output layer. Both transfer functions meet the standard that the majority of machine learning practitioners follow for predictive neural networks nowadays.

For the ANFIS, the chosen architecture is based on the Sugeno Fuzzy Inference System. Each input is attached to two membership functions (MF), resulting in 2^5 rules and output membership functions. The choice for the type of input membership function was also determined via for loop. The best fitting MF would automatically be chosen. For a fast and simple training, the output membership functions were chosen as constants instead of linear equations. This reduces significantly the number of parameters to update in backpropagation, making the algorithm fast.

For KNN, the input data is in a similar format to the ANN algorithm. It is a non-parametric method, meaning that it makes less assumptions and thus is a more robust method then parametric algorithms in some cases. They have been shown to be effective in predicting scenarios where there is limited amount of input data. We chose to use a standard Euclidean Distance algorithm for boundary detection and 5 neighbors, determined by square root of the number of training samples, i.e. $\sqrt{N = 31}$.

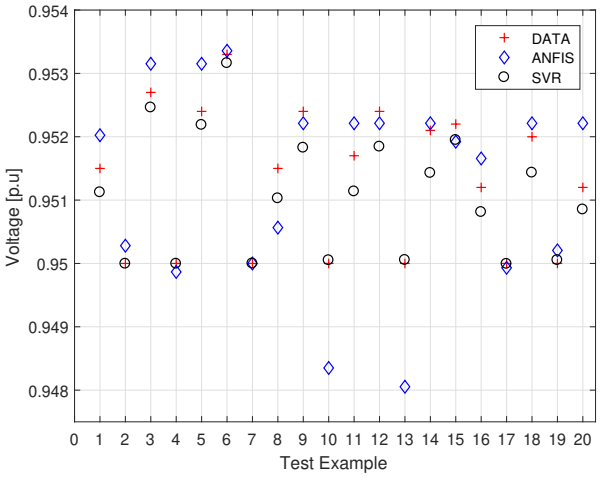
For SVR, the input data follows the same model used in ANN and KNN. It's characterized by the use of kernels, which in our model was a polynomial kernel. High value Regularization Parameters often misrepresent the output hypothesis due to its elevated degree in separating the support vectors. For this reason, we chose a small regularization value of $C = 0.5$ in order to avoid over-fitting.

And finally for DT, the input data follows the same standard used in all previous models except ANFIS due to its membership function inputs. Since the training data is small, we went forth with 3 folds, 1 seed for our root and 2 as our minimum number of instances.

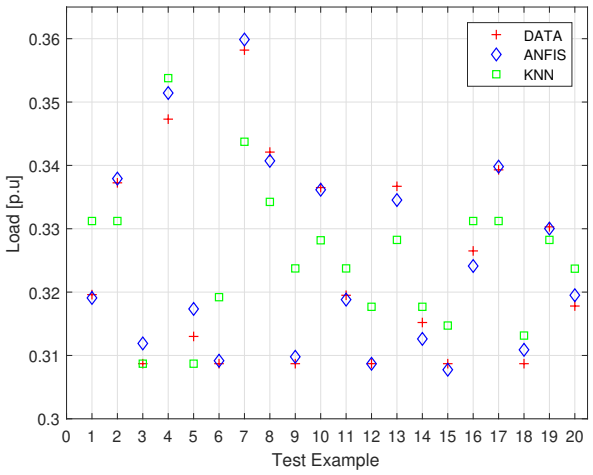
All prediction plots will contain the first twenty examples from the fifty example testing set, which was randomly selected. For a cleaner look, each plot will contain the two best predictions of all the regression algorithms together with the actual output test data.

5.5 Voltage and Load Mapping - First and Second Tests

The first two tests consist in comparing the critical voltage and load mapping hypothesis of the regression algorithms for unaltered input data as well as unaltered testing data, obtained from the continuation power flow simulation. Figure (17.a) is the comparison of ANFIS and SVR critical voltage prediction results in a side by side contrast with the output prediction of the load for ANFIS and KNN in Fig (17.b). It is worth re-mentioning that each plot compares the two best predictions out of the total amount of regression algorithms chosen in this work.



(a) Prediction comparison for critical voltage.



(b) Prediction comparison for load.

Figura 17 – Voltage and Load Mapping for unaltered data

Table 2 highlights the algorithms architecture as well as the root mean square error (RMSE) in tests 1 and 2. The RMSE evaluation method was chosen in order to plot a Radar Chart to rank the overall best regression algorithm among the five of them, shown in subsection 4.5 further below. Since the training set is very small, the training time is insignificant in all regression simulations conducted. For critical voltage mapping,

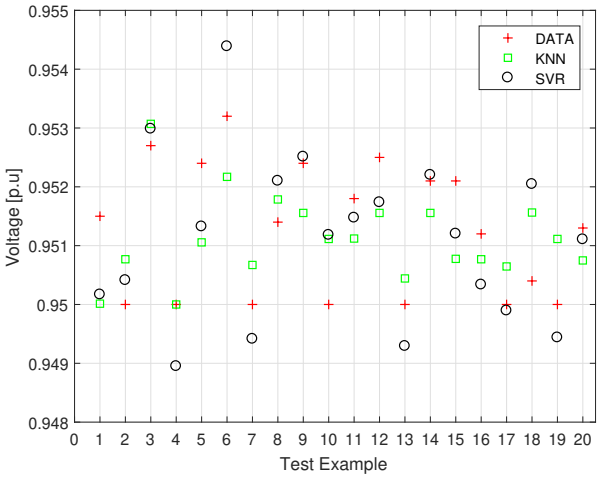
all algorithms show similar results when it comes to RMSE evaluation for prediction. On the other hand, ANFIS had a better performance in load mapping, when compared to the rest, shown by a 0.00189 value for RMSE.

Tabela 2 – RMSE comparison in unaltered data for ANFIS and ANN.

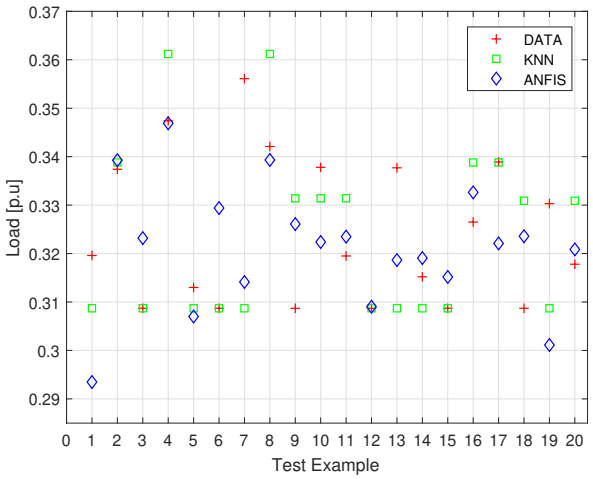
Critical Voltage				
ANFIS	ANN	KNN	SVR	DT
Batch Size = 31 Gauss MF Input Constant Output $2^5 rules$	Batch Size = 31 Data input ReLU output 1 hidden layer/ 3 neurons	Batch Size = 31 Data input Euclidean Boundary Neighbors: 5	Batch Size = 31 Data input PolyKernel Reg. Parameter (C = 0.5)	Batch Size = 31 # of Folds = 3 Seed randomize data: 1 Min. number of instances: 2
RMSE: 0.00061	RMSE: 0.00069	RMSE: 0.00075	RMSE: 0.00043	RMSE: 0.00064
Load				
ANFIS	ANN	KNN	SVR	DT
Batch Size = 31 Gauss MF Input Constant Output $2^5 rules$	Batch Size = 31 Data input ReLU output 1 hidden layer/ 3 neurons	Batch Size = 31 Data input Euclidean Boundary Neighboors: 5	Batch Size = 31 Data input PolyKernel Reg. Parameter (C = 0.5)	Batch Size = 31 # of Folds = 3 Seed randomize data: 1 Min. number of instances: 2
RMSE: 0.00189	RMSE: 0.01693	RMSE: 0.00751	RMSE: 0.01980	RMSE: 0.01120

5.6 Voltage and Load Mapping - Third and Forth Tests

The third and forth tests consist in comparing the critical voltage and load mapping hypothesis of the regression algorithms for unaltered training data but with a noisy testing data. This ensures that all techniques are trained with clean data while being tested with imprecise data, allowing to determine which method has a better interpolation property for the studied case. Both tests try to recreate what would happen when the machine learning algorithms are trained with simulation data while being tested with data coming from sensors in the field. The imprecision of the testing data is within a $\pm$ 5% range for all 50 examples. Figure (18.a) is the comparison of the two best predictive algorithm results, KNN and SVR, for the output of the critical voltage testing set. Meanwhile figure (18.b) is the comparison of the two best predictive algorithms results, ANFIS and KNN, for the output of the load testing set.



(a) Prediction comparison for critical voltage.



(b) Prediction comparison for load.

Figura 18 – Voltage and Load Mapping for noisy training data

Also Table 3 highlights the algorithms architecture as well as the RMSE in tests

3 and 4. For critical voltage mapping, all algorithms show similar results when it comes to RMSE evaluation.

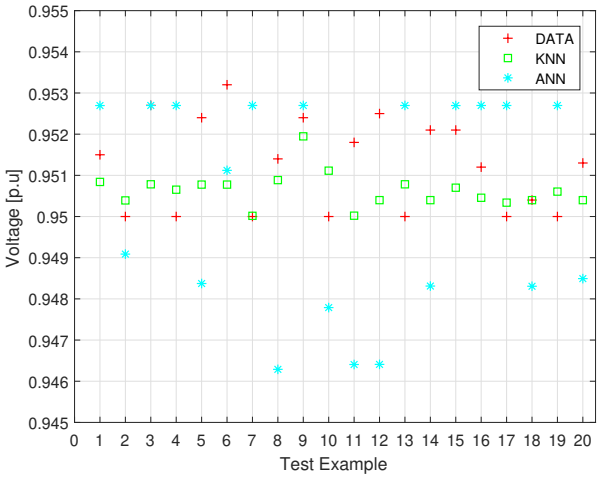
Contrarily, ANFIS has a better performance in load mapping with KNN being second best. An interesting remark from these tests is that KNN predictions are well balanced within the lower limit of the testing values. While other methods sometimes struggle to accurately predict the lowest values from the testing set, KNN is consistent and does not under-evaluate. For Critical Voltage mapping, SVR led the output prediction comparison with a RMSE of 0.00114 - although all regression algorithms aforementioned had similar predictions error. For Load mapping, ANFIS emerged as the best algorithm with a RMSE of 0.01660

Tabela 3 – RMSE comparison in noisy data for ANFIS and ANN.

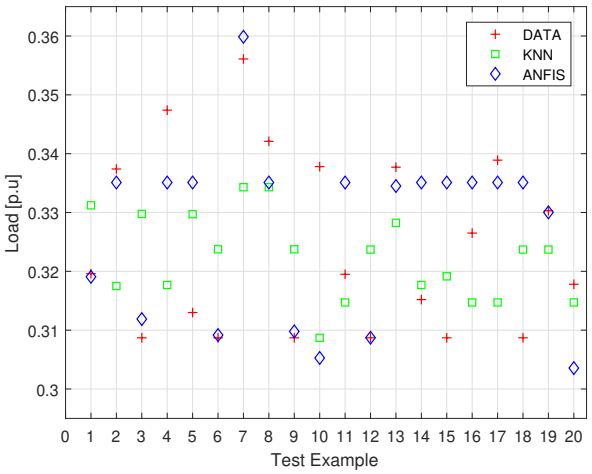
Critical Voltage				
ANFIS	ANN	KNN	SVR	DT
Batch Size = 31 Sigmoidal MF Input Constant Output $2^5 rules$	Batch Size = 31 Data input ReLU output 1 hidden layer/ 3 neurons	Batch Size = 31 Data input Euclidean Boundary Neighboors: 5	Batch Size = 31 Data input PolyKernel Reg. Parameter (C = 0.5)	Batch Size = 31 # of Folds = 3 Seed randomize data: 1 Min. number of instances: 2
RMSE: 0.00158	RMSE: 0.00141	RMSE: 0.00115	RMSE: 0.00114	RMSE: 0.00118
Load				
ANFIS	ANN	KNN	SVR	DT
Batch Size = 31 Gauss MF Input Constant Output $2^5 rules$	Batch Size = 31 Data input ReLU output 1 hidden layer/ 3 neurons	Batch Size = 31 Data input Euclidean Boundary Neighboors: 5	Batch Size = 31 Data input PolyKernel Reg. Parameter (C = 0.5)	Batch Size = 31 # of Folds = 3 Seed randomize data: 1 Min. number of instances: 2
RMSE: 0.01660	RMSE: 0.07102	RMSE: 0.38801	RMSE: 0.02010	RMSE: 0.02042

5.7 Voltage and Load Mapping - Fifth and Sixth Tests

The fifth and sixth tests consist in comparing the critical voltage and load mapping hypothesis of the aforementioned regression algorithms for unaltered training data but now with a lossy testing data. Similarly to the noisy testing set, the lossy testing set is used to compare the prediction capabilities of the machine learning algorithms when there is a loss of information from the sensors in the field. The gaps in data were randomly sorted for all fifty training examples. Both tests try to recreate what would happen when the different machine learning algorithms are trained with simulation data while being tested with very poor data coming from the field sensors. Figure (19.a) shows the comparison



(a) Prediction comparison for critical voltage.



(b) Prediction comparison for load.

Figura 19 – Voltage and Load Mapping for lossy testing data

of both KNN and ANN algorithm for critical voltage, which for this test showed to be the best fitting algorithms. Once again, KNN showed its most noticeable characteristic - it does not under-evaluate the minimum values from the testing set. Figure (19.b) shows the comparison of both KNN and ANFIS for load mapping, where ANFIS stood out with

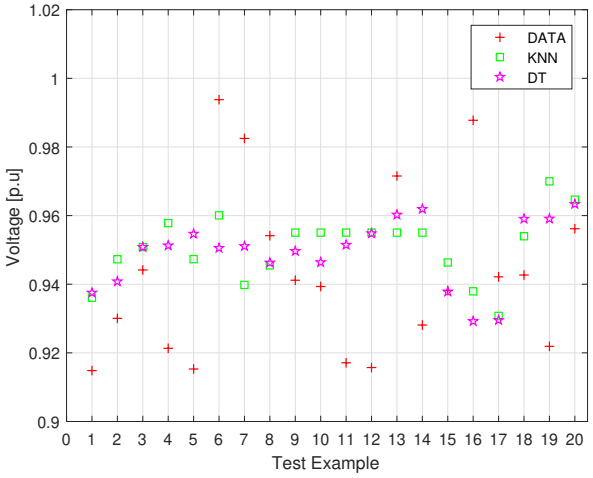
a RMSE of 0.01434. Table 4 highlights the algorithms architecture as well as the RMSE in tests 5 and 6.

Tabela 4 – RMSE comparison in lossy data for ANFIS and ANN.

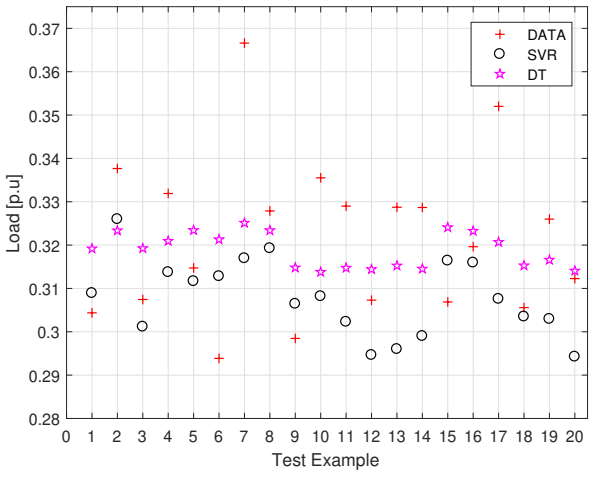
Critical Voltage				
ANFIS	ANN	KNN	SVR	DT
Batch Size = 31 Sigmoidal MF Input Constant Output $2^5 rules$	Batch Size = 31 Data input ReLU output 1 hidden layer/ 3 neurons	Batch Size = 31 Data input Euclidean Distance Boundary Number Neighboors: 5	Batch Size = 31 Data input PolyKernel Regularization Parameter (C = 0.5)	Batch Size = 31 # of Folds = 3 Seed randomize data: 1 Minimum number of instances: 2
RMSE: 0.00386	RMSE: 0.00290	RMSE: 0.00113	RMSE: 0.01033	RMSE: 0.00570
Load				
ANFIS	ANN	KNN	SVR	DT
Batch Size = 31 Gauss MF Input Constant Output $2^5 rules$	Batch Size = 31 Data input ReLU output 1 hidden layer/ 3 neurons	Batch Size = 31 Data input Euclidean Distance Boundary Number Neighboors: 5	Batch Size = 31 Data input PolyKernel Regularization Parameter (C = 0.5)	Batch Size = 31 # of Folds = 3 Seed randomize data: 1 Minimum number of instances: 2
RMSE: 0.01434	RMSE: 0.05890	RMSE: 0.01660	RMSE: 0.11440	RMSE: 4.72990

5.8 Voltage and Load Mapping - Seventh and Eighth Tests

The seventh and eighth tests compare the critical voltage and load mapping hypothesis of all the regression algorithms but with noisy training and testing data. The noisy training set is used to train the aforementioned algorithms in order to compare the prediction capabilities of the machine learning regression algorithms. Similarly to tests 3 and 4, the imprecision added to the training and testing data are between a ± 5% range. Both tests try to recreate what would happen when the different machine learning algorithms are trained and tested with noisy data coming from the field sensors. Figure (20.a) show the comparison of the two best predictive algorithms results for the output of the critical voltage testing set, while Fig. (20.b) exhibits the comparison of the two best predictive algorithms results for the output of the load testing set.



(a) Prediction comparison for critical voltage.



(b) Prediction comparison for load.

Figura 20 – Voltage and Load Mapping for noisy training and testing data

Table 5 puts in evidence the algorithms architecture as well as the RMSE for both tests 7 and 8. KNN had the better critical voltage prediction among all regression algorithms with a RMSE of 0.03110 while DT had the better load prediction with a RMSE of 0.01490.

Tabela 5 – RMSE comparison in lossy data for ANFIS and ANN.

Critical Voltage				
ANFIS	ANN	KNN	SVR	DT
Batch Size = 31 Sigmoidal MF Input Constant Output $2^5 rules$	Batch Size = 31 Data input ReLU output 1 hidden layer/ 3 neurons	Batch Size = 31 Data input Euclidean Boundary Neighbors: 5	Batch Size = 31 Data input PolyKernel Reg. Parameter (C = 0.5)	Batch Size = 31 # of Folds = 3 Seed randomize data: 1 Min. number of instances: 2
RMSE: 0.03620	RMSE: 0.05440	RMSE: 0.03110	RMSE: 0.03280	RMSE: 0.03140
Load				
ANFIS	ANN	KNN	SVR	DT
Batch Size = 31 Gauss MF Input Constant Output $2^5 rules$	Batch Size = 31 Data input ReLU output 1 hidden layer/ 3 neurons	Batch Size = 31 Data input Euclidean Boundary Neighboors: 5	Batch Size = 31 Data input PolyKernel Reg. Parameter (C = 0.5)	Batch Size = 31 # of Folds = 3 Seed randomize data: 1 Min. number of instances: 2
RMSE: 0.03863	RMSE: 0.03670	RMSE: 0.02750	RMSE: 0.01680	RMSE: 0.01490

5.9 Overall Regression Score

All of the regression algorithms chosen for this study were thoroughly tested in subsections 4.1 to 4.4. We were able to evaluate their efficiency based on the RMSE, but despite many tests, it is unclear as to how we can rank the precision and prediction capabilities of the five regression algorithms with the RMSE as it stands.

With this in mind, we propose a scoring method that ranks the overall efficiency of the algorithms from tests 1 to 8. The score is the total sum of every normalized inverse RMSE, meaning that an unit score is the best grade an algorithm can achieve while null being the worst grade. Table 6 shows the score for each algorithm based on the tests, going from test 1 to 8. A grand total score is then shown at the end of the table, where the highest number points to the overall best algorithm and the lowest number points to the worst overall algorithm. It is possible to conclude that both ANFIS and KNN had three unit grades and that their total sum was approximately 5.93 and 6.02. All other algorithms weren't remotely close to their grand total score, meaning that out of the five algorithms, ANFIS and KNN were clearly superior. KNN has a marginal advantage compared to ANFIS, although the testing data predictions from ANFIS could be filtered in order to eliminate the under-evaluated minimum values, which would make it even more precise.

Tabela 6 – Overall prediction score.

TEST / ALGORITHM	ANFIS	ANN	SVR	KNN	DT
1. UNALTERED VOLTAGE	0.698908197	0.617875362	1	0.5329175	0.66505577
2. UNALTERED LOAD	1	0.11181787	0.095440505	0.251962933	0.16872518
3. 5% TRAINING VOLTAGE	0.723053875	0.808510638	0.991304348	1	0.96610169
4. 5% TRAINING LOAD	1	0.233680245	0.042772019	0.82979855	0.81273119
5. LOSSY VOLTAGE	0.259081587	0.344827586	0.096805421	1	0.1754386
6. LOSSY LOAD	1	0.243478947	0.125357605	0.863910241	0.00303197
7. 5% TRAIN & TEST VOLTAGE	0.859116022	0.571691176	0.948170732	1	0.99044586
8. 5% TRAIN & TEST LOAD	0.385728841	0.40599455	0.886904762	0.541818182	1
Grand Total	5.925888521	3.337876376	4.186755391	6.020407406	4.78153026

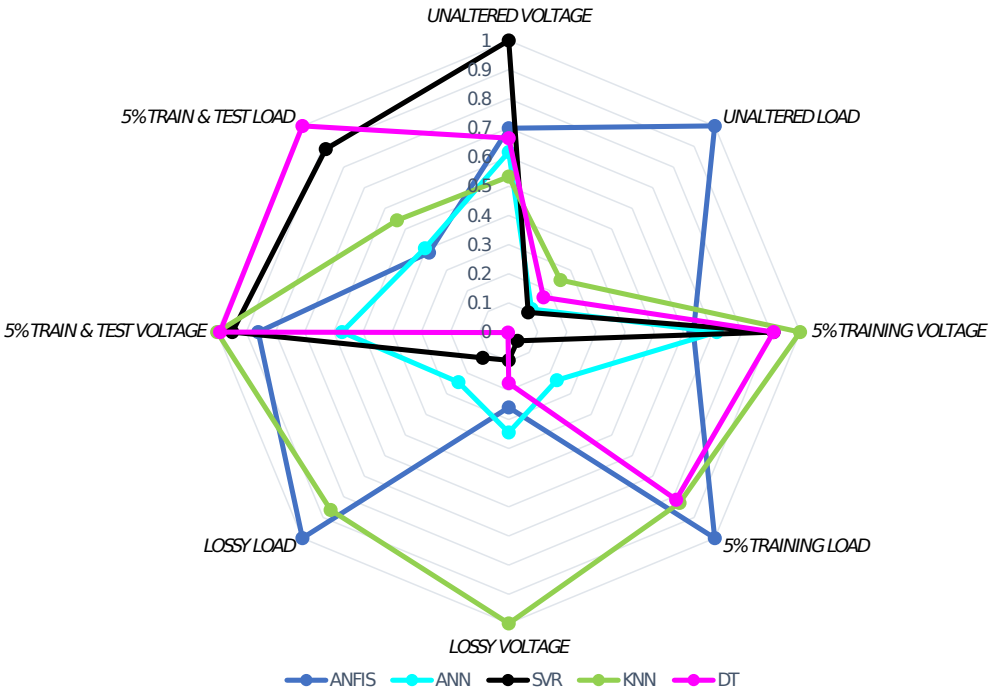


Figura 21 – Radar chart of the regression algorithms for each test.

Figure (21) is the radar chart of results of each algorithm per test, starting with *unaltered voltage* (test 1) and ending at 5% *train & test load* (test 8). Each color matches the algorithms that were labeled in the figures from subsections 4.1 to 4.4.

6 Conclusion

Based on the present analyses, ANFIS and KNN have a better performance in critical voltage and load prediction, within the VLHA hatched area, when compared to SVR, DT and ANN. One of the reasons why ANFIS had a better prediction capability is that the input membership functions serve as additional input data for the training process. Also, the simplest ANFIS model with a constant output membership function rather than a linear one, produces good enough predictions with only two training epochs. KNN, on the other hand, also proved to work well in situations where training data is scarce.

Although KNN and ANFIS had similar overall results, there is a clear advantage on using ANFIS. The added benefit of using ANFIS is having the rules and the membership functions that map the input to output relationship. These rules and membership functions allow a simple assessment tool for critical voltage and load, ideal for moments when there is the need to generate fast results of different scenarios of the system. For greater systems in which a continuation power flow algorithm can take up several minutes to generate all PV curves for increasing loads, the ANFIS assessment tool takes a couple of seconds to run and generate outputs for the desired analysis.

In future work, machine learning algorithms will be used to generate a critical voltage classification tool for different system loading scenarios as well as implement a robust PCA algorithm for voltage area division near the critical point in high dimensional power systems. We will also implement an online critical voltage mapping setup using ANFIS, since it showed to be practical and gives good estimation of the voltage and load levels.

7 Published Papers

This work was supported in part by the Brazilian Funding Agencies CAPES, CNPq, FAPEMIG and INERGE. It resulted in the following published papers:

- FACHINI, F.; LOPES, B. I. L. Critical Bus Voltage Mapping using ANFIS with regards to Max Reactive Power in PV buses. In: 2019 IEEE Milan PowerTech. IEEE, 2019. p. 1-6.
- FACHINI, F.; FULY, B. I. L. A Comparison of machine learning regression models for critical bus voltage and load mapping with regards to max reactive power in PV buses. Electric Power Systems Research, v. 191, p. 106883.

Referências

1 CHAKRAVORTI, T. et al. A new robust kernel ridge regression classifier for islanding and power quality disturbances in a multi distributed generation based microgrid. *Renewable Energy Focus*, Elsevier, v. 28, p. 78–99, 2019. 16

2 REUTERS. *UPDATE 2-Blackout on Indonesia's Java due to coal disruption.* 2008. <https://www.reuters.com/article/power-indonesia-idUSSP17318520080221>. 16

3 BILLIONS, C. E. for. *Indian Blackouts of July 2012: What Happened and Why?* 2016. <https://medium.com/clean-energy-for-billions/indian-blackouts-of-july-2012-what-happened-and-why-639e31fb52ad>. 16

4 SINGH, B.; SHARMA, N.; TIWARI, A. Prevention of voltage instability by using facts controllers in power systems: a literature survey. *International Journal of Engineering Science and Technology*, v. 2, n. 5, p. 980–992, 2010. 17

5 DONG, F. et al. Cause and effects of voltage collapse-case studies with dynamic simulations. In: IEEE. *IEEE Power Engineering Society General Meeting, 2004.* [S.l.], 2004. p. 1806–1812. 17

6 AJJARAPU, V.; CHRISTY, C. The continuation power flow: a tool for steady state voltage stability analysis. In: IEEE. *[Proceedings] Conference Papers 1991 Power Industry Computer Application Conference.* [S.l.], 1991. p. 304–311. 17, 22, 26

7 CHAKRAVORTY, M.; DAS, D. Voltage stability analysis of radial distribution networks. *International Journal of Electrical Power & Energy Systems*, Elsevier, v. 23, n. 2, p. 129–135, 2001. 17

8 KESSEL, P.; GLAVITSCH, H. Estimating the voltage stability of a power system. *IEEE Transactions on power delivery*, IEEE, v. 1, n. 3, p. 346–354, 1986. 17

9 MORISON, G.; GAO, B.; KUNDUR, P. Voltage stability analysis using static and dynamic approaches. *IEEE transactions on Power Systems*, IEEE, v. 8, n. 3, p. 1159–1171, 1993. 17

10 MALBASA, V. et al. Voltage stability prediction using active machine learning. *IEEE Transactions on Smart Grid*, IEEE, v. 8, n. 6, p. 3117–3124, 2017. 18

11 ZHOU, D. Q.; ANNAKKAGE, U. D.; RAJAPAKSE, A. D. Online monitoring of voltage stability margin using an artificial neural network. *IEEE Transactions on Power Systems*, IEEE, v. 25, n. 3, p. 1566–1574, 2010. 18, 19

12 AJJARAPU, V.; CHRISTY, C. The continuation power flow: a tool for steady state voltage stability analysis. *IEEE transactions on Power Systems*, Ieee, v. 7, n. 1, p. 416–423, 1992. 18

13 SAJAN, K.; KUMAR, V.; TYAGI, B. Genetic algorithm based support vector machine for on-line voltage stability monitoring. *International Journal of Electrical Power & Energy Systems*, Elsevier, v. 73, p. 200–208, 2015. 19

14 ZHANG, R. et al. Voltage stability margin prediction by ensemble based extreme learning machine. In: IEEE. *2013 IEEE Power & Energy Society General Meeting*. [S.l.], 2013. p. 1–5. 19

15 CAI, H.; MA, H.; HILL, D. A data-based learning and control method for long-term voltage stability. *IEEE Transactions on Power Systems*, IEEE, 2020. 19

16 LI, H.; CHEN, C. P.; HUANG, H.-P. *Fuzzy neural intelligent systems: Mathematical foundation and the applications in engineering*. [S.l.]: CRC Press, 2000. 20, 48

17 HAYKIN, S. *Neural networks*. [S.l.]: Prentice hall New York, 1994. v. 2. 20, 48

18 FACHINI, F.; LOPES, B. Critical bus voltage mapping using anfis with regards to max reactive power in pv buses. In: IEEE. *2019 IEEE Milan PowerTech*. [S.l.], 2019. p. 1–6. 20, 27, 29, 47

19 AWAD, M.; KHANNA, R. Support vector regression. In: *Efficient learning machines*. [S.l.]: Springer, 2015. p. 67–80. 20

20 SANTOS, M. et al. Secondary voltage control system based on fuzzy logic. *Electric Power Systems Research*, Elsevier, v. 119, p. 377–384, 2015. 22, 25, 26

21 ZHONG, J. et al. Localized reactive power markets using the concept of voltage control areas. *IEEE Transactions on Power Systems*, IEEE, v. 19, n. 3, p. 1555–1561, 2004. 22

22 JOLLIFFE, I. *Principal component analysis*. [S.l.]: Springer, 2011. 22

23 ABDI, H.; WILLIAMS, L. J. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, Wiley Online Library, v. 2, n. 4, p. 433–459, 2010. 22

24 NG, A. *Machine Learning*. 2019. <https://www.coursera.org/learn/machine-learning>. 23, 31

25 SHLENS, J. A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*, 2014. 23

26 MONTICELLI, A. J. *Fluxo de carga em redes de energia elétrica*. [S.l.]: E. Blucher, 1983. 25, 36

27 GAO, B.; MORISON, G.; KUNDUR, P. Voltage stability evaluation using modal analysis. *IEEE transactions on power systems*, IEEE, v. 7, n. 4, p. 1529–1542, 1992. 26

28 MICHIE, D. et al. Machine learning. *Neural and Statistical Classification*, Technometrics, v. 13, n. 1994, p. 1–298, 1994. 29

29 AWAD, M.; KHANNA, R. *Efficient learning machines: theories, concepts, and applications for engineers and system designers*. [S.l.]: Springer Nature, 2015. 32

30 HAYKIN, S. *Neural networks: a comprehensive foundation*. [S.l.]: Prentice-Hall, Inc., 2007. 32

31 ŞAHIN, M.; EROL, R. A comparative study of neural networks and anfis for forecasting attendance rate of soccer games. *Mathematical and Computational Applications*, Multidisciplinary Digital Publishing Institute, v. 22, n. 4, p. 43, 2017. 34

32 LIN, C.-T.; LEE, C. G. Neural fuzzy systems. *PTR Prentice Hall*, 1996. 34

33 CZOGALA, E.; LESKI, J. *Fuzzy and neuro-fuzzy intelligent systems*. [S.l.]: Physica, 2012. v. 47. 34

34 JANG, J.-S. R. et al. Fuzzy modeling using generalized neural networks and kalman filter algorithm. In: *AAAI*. [S.l.: s.n.], 1991. v. 91, p. 762–767. 34

35 HINES, J.; TSOUKALAS, L. H.; UHRIG, R. E. *MATLAB supplement to fuzzy and neural approaches in engineering*. [S.l.]: John Wiley & Sons, Inc., 1997. 34, 48

36 JANG, J.-S. Anfis: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics*, IEEE, v. 23, n. 3, p. 665–685, 1993. 34

37 TSOUKALAS, L. H.; UHRIG, R. E. Fuzzy and neural approaches in engineering. John Wiley, 1997. 35

38 AJJARAPU, V.; CHRISTY, C. The continuation power flow: a tool for steady state voltage stability analysis. *IEEE transactions on Power Systems*, IEEE, v. 7, n. 1, p. 416–423, 1992. 37

39 AJJARAPU, V. *Computational techniques for voltage stability assessment and control*. [S.l.]: Springer Science & Business Media, 2007. 38

40 CYBENKO, G. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, Springer, v. 2, n. 4, p. 303–314, 1989. 49