Universidad de Alcalá
Escuela Politécnica Superior

Master Universitario en Ingeniería Industrial

Trabajo Fin de Máster

Detección de fallos en el equipamiento de medida en los centros de transformación

Autor: Javier Moriano Martin

Tutor: Francisco Javier Rodríguez Sánchez

2016

UNIVERSIDAD DE ALCALÁ

ESCUELA POLITÉCNICA SUPERIOR

Master Universitario en Ingeniería Industrial

Trabajo Fin de Máster

Detección de fallos en el equipamiento de medida en los centros de transformación

Autor: Javier Moriano Martin

Director: Francisco Javier Rodríguez Sánchez

Tribunal:

Presidente: Emilio Jose Bueno Peña

Vocal 1º: Jesus Sánchez Golmayo

Vocal 2º: Francisco Javier Rodríguez Sánchez

Calificación: .....

Fecha: .....

A mis padres y a mi hermano
por su apoyo continuo.

"Cuanto más atrás puedas mirar, más adelante verás."
Winston Churchill

Acknowledgements

En este trabajo fin de master quiero dar las gracias a todas las personas que me han acompañado durante mi etapa universitaria. En primer lugar a mis compañeros de universidad, con los que tantas horas he compartido y en especial a aquellos con los que las horas de clase se hacían insuficientes.

En segundo lugar a los profesores que han conseguido aportarme los conocimientos necesarios y a los que han sido un referente como ingenieros y como personas. Especialmente a Pedro Martín, Emilio Bueno y a mi tutor Francisco Javier por confiar en mí y darme esta oportunidad.

Por último, a todas aquellas personas importantes en mi vida. A mis amigos por todos los grandes momentos compartidos y a Laura por intentar entender e interesarse por mis historias de ingeniería. En especial agradecer a mi familia por darme el apoyo en todos los sentidos para permitirme formarme y llegar a completar mis objetivos.

Quien piensa poco, se equivoca mucho.

Leonardo Da Vinci

Resumen

En los últimos años, a los centros de transformación se les está proporcionando un equipamiento que permite la gestión completa del mismo. De este modo, de producirse medidas erróneas podría ser muy perjudicial para el normal funcionamiento del centro.

En este trabajo, empleando las técnicas de predicción de carga que permiten estimar cual será la carga eléctrica en el futuro, será posible detectar dos errores sistemáticos de medidas diferentes.

Para la predicción de carga, se utilizará una red neuronal y el análisis de componentes principales será empleado para comprimir la información.

Finalmente, el sistema desarrollado será probado con datos obtenidos del simulador implementado por un distribuidor eléctrico y datos reales proporcionados por una compañía eléctrica.

Palabras clave: Predicción de Carga, Detección de Errores de Medida, Red Neuronal, Análisis de Componentes Principales, Centro de Transformación.

Abstract

In recent years, Secondary Transformer Substations are being provided with equipment to allow the full management of the substation. In this way, erroneous measurements could be very harmful for the normal performance of the substation.

In this work, employing Short Term Load Forecasting (STLF) which allows to estimate what the future electric load will be, it is possible to detect two different systematic measurements errors. For the load forecasting, the chosen system is composed by an artificial neural network and principal component analysis is employed for information compression.

Finally, developed system is tested with simulated data obtained from a simulator provided by a Transmission System Operator and real data provided by an Electrical Company. Different levels of gain and offset errors are detected.

Keywords: Short Term Load Forecasting, Measurement Error Detection, Artificial Neural Network, Principal Component Analysis, Secondary Substation.

Extended Abstract

In last years, Secondary Transformer Substations are being equipped with different devices to head substations toward an autonomous management environment. It is a powerful enhancement for the secondary substations due to the maintenance simplification. However, it implies a high risk due to the acquired measurements are employed for acting and an erroneous measurement could imply a danger in the normal performance of the operation. In this context, erroneous measurement detection appears as a powerful tool to prevent the secondary substation well performance.

The electronic instrumental transducers are the devices employed for acquiring the measurements and the ones that could start taking erroneous measurement at any moment. The erroneous measurements could be produced for a failure at the device or for device ageing. Usually a failure at the device implies erroneous measurements that are easily detectable. However, there exists other kind of failure at the electronic instrumental transducers that are difficult to detect. These systematic errors, namely, offset and gain errors are harder to identify. Their detection in a reasonable time could result in an interesting information for the electrical companies which could identify which device is failing and send an operator to accomplish a field test and replace the device if needed.

The measurements collected at a secondary substation, presents a high degree of periodicity depending on the day of the week and it varies depending on the month. It makes possible that the power consumption (directly related with the current consumption) for a day can be obtained as a relationship between the load consumption of its historical measurements and the day of the week and month of the year. The historical measurement set is made up of the hourly loads for the previous day, the ones for the same day one week before and the ones for the same day two weeks before.

As the historical data set of each day to forecast presents 72 measurements and those measurements are highly correlated, compression techniques can be employed for reducing the dimensionality which would decrease the processing time in the rest of the project and would reduce the amount of information. With this information, employing different techniques it is possible to forecast the measurements.

Different techniques for forecasting are compared. Firstly, artificial neural networks, which is based on the neural processing commonly found in the human brain is studied. The artificial neural network is employed with the historical compressed information as input and it is trained with known forecast measurements. A particular artificial neural network is needed for each electronic instrumental transducer to be studied. The results obtained employing artificial neural networks are also compared with the ones obtained with two hybrid models compounds by the combination of artificial neural networks and singular spectrum analysis.

Singular spectrum analysis consists on a powerful technique employed in the field of time series analysis. This method consists on two stages: decomposition and reconstruction. In this method the original series are decomposed in different components and then the time series is reconstructed with the chosen components eliminating the noise. This technique can be employed as a filter before the artificial

neural network (Hybrid method 1) or a different neural network can be employed for each principal component and then with the reconstruction of the output of each artificial neural network the forecast load are obtained (Hybrid method 2). It would gives two different forecast load concurrently which can be combined according to the accuracy of each method by a weighted average taking the best of each method.

Once the forecast load are obtained independently on the employed method forecasting method, the forecast load are compared with the real measurements token by the electronic instrumental transducer. The novel method for comparison described in this work can independently identify the level of gain and offset error. It is possible due to the fact that the differentiation in both measured and forecast measurements would only be affected by a gain measurements and once that the gain measurement error is obtained, the offset error can be obtained. In this way a system that is able to independently detect the level of gain and offset error can be implemented.

Then, the system needs to be tested against collected measurement data. For this purpose, a simulator developed by a transmission system operator generates simulated measurements for a chosen period of time. Therefore, employing a period of two years data, the data set is enough for being divided in three group. First group is employed for training the forecast system. The second one two validate it and the the third one to test it. All the process is carried out over MATLAB R2014a and once the forecast process system is correctly trained, errors are manually injected over the next set of measurements. Different level of gain and offset errors are injected, in some cases simultaneously and they are successfully detected.

Finally, the system is tested with real data that was provided from different real Secondary Transformer Substations. The quality of this data is not as good as the one obtained from the simulator but it is possible to demonstrate how our system, by employing different averaging processes, is able to obtain a good approximation to the real levels of gain and offset errors.

The described system developed during this work, has also been published in an open access journal. Sensors - Open Access Journal (ISSN 1424-8220; CODEN: SENSC9), which belongs to the first quartile in Instruments and Instrumentation area, was the chosen journal for publication. Sensors is the leading international, peer-reviewed, open access journal on the science and technology of sensors and biosensors.

Contents

List of Figures

List of Tables

List of Acronyms

ANN	Artificial Neural Network.
EIT	Electronic Instrumental Transducer.
MAPE	Mean Absolute Percentage Error.
MLP	Multilayer Perceptron.
MSE	Mean Square Error.
NMBE	Normalized Mean Bias Error
NRMSE	Normalized Root Mean Square Error
PCA	Principal Component Analysis.
PCs	Principal Components.
SAS	Substation Automation System.
SOM	Self Organizing Map.
SSA	Singular Spectrum Analysis.
STLF	Short Term Load Forecasting.
SVD	Singular Vector Decomposition.

Chapter 1

Introduction

1.1. Presentation

With the development of communication and information technology over past decades, broadband communication network has been prevalent in power systems. As a protocol to regulate communication within *Substation Automation System (SAS)*, IEC 61850 is becoming popular in recent years [6].

Unlike traditional SAS where Intelligent Electronic Devices (IEDs) are hardwired linked to implement data acquisition and carry out their function, development of Electronic Instrumental Transducer *Electronic Instrumental Transducer (EIT)* and prevalence of communication and information technology have led to a revolution in SAS using this new protocol. In the network based SAS, current and voltage are measured with EIT and the output is sent to secondary equipments as numerical signals via broadband communication networks. Thereafter, the numerical measurements can be utilized conveniently for controlling and protection applications of SAS.

However, as two edges of a sword, the risk emerges is erroneous measurement that may be introduced in the signal (voltage and current) acquisition and measurement transmission. Erroneous measurement received by protection system may lead to its mis-operation. Generally speaking, the major causes are mal-function of EIT and broadband communication network.

The voltages and current consumption are very related with their historical measurements data depending on different factors. From the earliest times, it has always been a way to enable the physical balance between the supply and the demand, allowing a reliable system operation. In this way, load forecast is an antique issue in the electricity sector domain and many different load forecasting models have been proposed during the last years.

In this work, by means of the load forecast techniques and comparing the forecast measurements with the real ones, different systematic error levels can be estimated. Concretely, this work is focused on gain and offset error detections and our developed system is tested against both simulated and real secondary substations data.

1.2. Motivation and targets

The motivation of this work is to develop a system which is able to detect measurement errors in a way that differs from the current ones. This system allows to control from the same place if any EIT is not measuring properly and in this way, an exhaustive test could be carried out over this device. Therefore, this system enhances the working of secondary substations in an economical way.

The main goals of this work are divided in two:

- Develop a system able to estimate the offset measurement error level of different EITs.
- Develop a system able to estimate the gain measurement error level of different EIT.

At the same time different secondary goals related with the main ones need to be satisfied as well:

- Basing on the current literature, choose the load forecasting model that best fits our necessities and implement it.
- Test the system against simulated and real secondary substation data.
- Develop a graphical user interface which allows the user to easily obtain the error levels.

1.3. Project report structure

This project report is divided in the following sections, firstly in chapter 2 the principles for the development work are explained and the utilised tools are described. Then, in chapter 3, the developed system is detailed and the principal features are given. Finally, in chapter 4 the results of testing the developed system against simulated and real data are provided.

Chapter 2

Theoretical study

2.1. Introduction

This work employs different tools for achieving the described goals. These tools are explained during this chapter and the different principles that this work is based on are detailed.

An introduction to each of the employed tools and principles are given and some references are also provided if greater detail is required.

This chapter starts with the description of the state of the art where the work which are the most relevant to this project are described and the most important features of each work are highlighted.

Then, the techniques and principles in which this project is based are described and some references are given in case than a more exhaustive explanation is needed.

2.2. State of the art

In the past few decades, significant advances in communication and information technology have accelerated the development and introduction of new broadband communication technologies in power systems. This fact has facilitated the power system automation in substations and SSs. With the aim of taking advantage of the current technology, there exists the IEC 61850 global standard for substation automation which is becoming particularly popular in recent years [6].

Nowadays, the development of Electronic Instrumental Transducer (EIT) and the importance of communication and information technology are quickly gaining acceptance in SSs. Consequently, traditional Substation Automation System (SAS) where Intelligent Electronic Devices (IEDs) were hardwired linked are being challenged by this new protocol standardized as IEC 61850. Different EITs carry out the necessary current and voltage measurements sent as numerical signals, which are used to control and protect the SSs. However, the advantages that this approach brings, could be negatively affected by the existence of erroneous measurements in the process of the signal acquisition. This could lead to inefficient operation of the protection system. [7].

On the other hand, the load profile and therefore the current consumption are strongly related to their historical measurement data depending on different data depending on different variables, such as economic factors and environmental data. With the aim of continuously maintaining the electricity generation and consumption balance, several approaches to load forecasting have been proposed during the last few years and some good examples can be found in [8] and [9].

In this work, by using load forecasting techniques and comparing the forecast loads with the real ones, different systematic error levels can be estimated, namely: gain and offset errors. The implemented system is tested against data from both a SS simulator and electricity suppliers.

The scope of this work falls within two research fields: (1) short term load forecasting; and (2) erroneous and false measurement detection. With this in mind, what follows presents the literature on both fields.

As far as load forecasting is concerned, in literature various approaches have been proposed to solve this issue. Load forecasting involves estimating the future electric load, for a forecast horizon, based on the available information about the state of the system.

In this regard, this work focuses on short term load forecast (STLF) to estimate the load with a horizon from one hour to one week. In [10] authors propose a new neural network approach to STLF based on a new modified learning algorithm. The results obtained are compared with those from other works [11][12] with the aim of evaluating the validity of the proposed approach. Despite of in some works [13][14] environmental variables are also employed, in others [15][16] only the historical load consumption is considered without a detrimental impact on the accuracy. The latter is the strategy adopted in this book.

In [15] the authors propose a solution for STLF in microgrids based on a three-stage architecture, which consists of a Self Organizing Map (SOM), a clustering via k-means algorithm, and a Multilayer Perceptron (MLP). A set of 29 inputs are provided to the MLP: 24 with the hourly load of the previous day; 2 for the day of the week of the previous day and 2 for the month of the previous day in the form of sines and cosines; and lastly, one more input that represents the next day total load estimation is provided. Finally, a total of 24 outputs corresponding to the estimated load for de forecasting day (d) are obtained.

In [16] a different approach to one-day ahead load profile prediction is presented. Employing an Artificial Neural Network (ANN) and testing three different scenarios with different inputs, the mean absolute percentage of the forecasting error is reduced between 0.5 % and 16 %, depending on the nature of the concurrent methodology used and the forecast day. The input information of the ANN depends on the model implemented. Model I requires 72 inputs, namely: hourly load values of the last day available, load values of the same weekday of previous week and load values of the same weekday 2 weeks before. Adding three more inputs, 2 for the day of the week and 1 for normal or vacation, the authors define a second model (Model II). In both models, 24 outputs are obtained which form the hourly predicted load for the forecast day. These outputs and the 72 inputs of the first model could also be used as input of a new ANN with a total of 96 inputs (Model III).

Regarding the topic of erroneous measurements detection, this problem has been addressed in different papers over several topics, as in [17] where the sensor failure detection in process control systems is discussed employing a neural network. In [18] a method based on fractal dimension is implemented for sensor fault diagnosis. In [19] another neural-network-based algorithm is employed for sensor failure detection for a flight control system.

Focusing in erroneous measurements in SSs, in [20] pattern identification is utilised to detect erroneous measurements. In order to achieve high pattern identification precision within the time limit imposed by the protection systems, a Radial Basis Function Neural Network (RBFNN) and an Orthogonal Least Square (OLS) learning algorithm are implemented. For detecting these errors, measurements from all the EITs are provided to the neural network as input and the only output of the neural network shows the probability of fault occurrence.

2.3. Utilized techniques

In this section, the most important tools that were employed in this work will be described. Firstly, in 2.3.1, it will be given a general vision of artificial neural networks which were used for the load forecasting. Then, in 2.3.2 the statistical procedure of Principal Components Analysis which allows to compress the information when the data is highly correlated is explained. Another method which can be employed in forecasting , the *Singular Spectrum Analysis (SSA)* spectral estimation method is also described. Finally, in 2.3.4, the description for the measurement errors is detailed.

2.3.1. Artificial Neural Network

In this subsection an overview of *Artificial Neural Network (ANN)* is given. The ANN, based on the neural processing commonly found in the human brain, and its learning ability, has been widely used in last years and it has been gaining more enthusiasts.

This technique is particularly interesting in problem categories that cannot be formulated as an algorithm, moreover, in problems where many factors affect. [1]

On the other hand, a disadvantage of this distributed fault tolerant storage is the fact that at first sight it cannot be realized what a neural network knows and performs or where its faults lie.

In this way, the main characteristics that are tried to adapt from biology to ANN are the following ones:

- Self-organization and learning capability
- Generalization capability
- Fault tolerance

A technical neural network consists of simple processing units, the neurons, and directed, weighted connections between those neurons. The strength of a connection (or the connecting weight) between two neurons i and j is referred to as $w_{i,j}$.

A neural network is a sorted triple (N, V, w) with two sets N, V and a function w, where N is the set of neurons and V a set $\{(i, j) | \ i, j \in \mathbb{N}\}$ whose elements are called connections between neuron i and neuron j.

The weights can be implemented in a square weight matrix W with the row number of the matrix indicating where the connection begins, and the column number of the matrix indicating, which neuron is the target.

For a neuron j, the propagation function receives the outputs $o_{i1}, \ldots, o_{in}$ of other neurons $i_1, i_2, \ldots,$ in (which are connected to j), and transforms them in consideration of the connecting weights $w_{i,j}$ into the network input net_j that can be further processed by the activation function 2.1. Thus, the network input is the result of the propagation function as can be seen in figure 2.1.

$$net_j = \sum_{i \in I} (o_i \cdot w_{i,j}) \tag{2.1}$$

Every neuron is, to a certain extent, at all times active. The reactions of the neurons to the input values depend on this activation state. The activation state indicates the extent of a neuron's activation

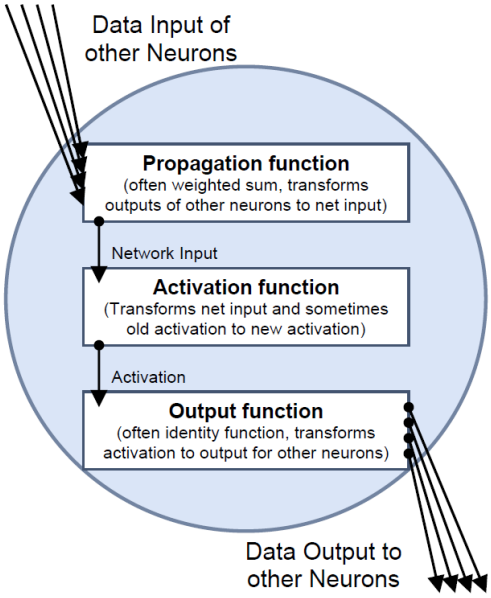


Figure 2.1: Data processing of a neuron. [1]

and is often shortly referred to as activation. The threshold value represents the threshold at which a neuron starts firing.

Finally, the output function of a neuron j calculates the values which are transferred to the other neurons connected to j.

The neural network working can be seen as a system, there are different vectors that need to be described: Firstly, the input vector x, which can be entered into the neural network. Then, depending on the type of network being used the neural network outputs an output vector y. Finally, the training sample p, which basically is an input vector which is used for training purposes because the desired output vector t is known.

The learning strategy is an algorithm that can be used to change and thereby train the neural network, so that the network produces a desired output for a given input.

Regarding to learning, a first classification can be done depending on the learning way:

- Unsupervised learning: Only the input patterns are given; the network tries to identify similar patterns and to classify them into similar categories.
- Supervised learning: the training set consists of input patterns as well as their correct results in the form of the precise activation of all output neurons.

Then, Another classification according to if the learning has any kind of feedback can be carried out:

- Offline learning: Several training patterns are entered into the network at once.
- Online learning: The network learns directly from the errors of each training sample.

Therefore, the best learning way needs to be selected depending on the desired application and the information that is available. In this way, if a classification is needed, an unsupervised learning suits better. In the case that a feedback could be interesting, where the system needs to adapt continuously to new situations, an online learning is the best option.

In this project, as the real measurements are available the day after, a supervised learning is used for the load forecast. Furthermore, an online learning is chosen, where the system adapts to possible changes in different fields that can affect the load demand such as economical or electricity prices.

In addition to the learning strategy, another factor that influences the working way is the algorithm used for the ANN optimization. Between the different options of neural network topologies, there exists several algorithms that can face the situation described in this work. In this case Levenberg-Marquardt backpropagation alogrithm was selected:

Levenberg-Marquardt is a network training function that updates weight and bias values according to Levenberg-Marquardt optimization. Levenberg-Marquardt is often the fastest backpropagation algorithm, and is highly recommended as a first-choice supervised algorithm, although it does require more memory than other algorithms.

Levenberg-Marquardt Optimization is a virtual standard in nonlinear optimization which significantly outperforms gradient descent and conjugate gradient methods for medium sized problems. It is a pseudo-second order method which means that it works with only function evaluations and gradient information but it estimates the Hessian matrix using the sum of outer products of the gradients. [21]

The Levenberg-Maquardt algorithm has been chosen for the ANN training two reasons:(i) the number of iterations required for the ANN training is lower in comparison to other techniques,(ii) this algorithm always guarantees the convergence of the training process[22][23].

More information about the employed ANN in this work can be found in next chapter, in section 3.3.2. There the employed inputs, the outputs that are obtained and the number of neurons that were used are explained.

2.3.2. Principal Component Analysis

The central idea of principal component *Principal Component Analysis (PCA)* is to reduce the dimensionality of a data set consisting of a large number of interrelated variables, while retaining as much as possible of the variation present in the data set. This is achieved by transforming to a new set of variables, the *Principal Components (PCs)*, which are uncorrelated, and which are ordered so that the first few retain most of the variation present in all of the original variables. [24]

PCA is a way of identifying patterns in data, and expressing the data in such a way as to highlight their similarities and differences. Once these patterns are found in the data, it can be compressed, i.e. by reducing the number of dimensions, without much loss of information [25]. This technique has been successfully employed in different fields such as face recognition [26], voice processing [27] and in ultrasonic sensor [3].

For the explanation, an scenario with two dimensions data will be supposed and the reason why this have been chosen is because in this way different plots of the data to show what the PCA analysis is doing at each step can be provided.

Step 1: Subtract the mean

For PCA to work properly, it is needed to subtract the mean from each of the data dimensions. In this way, a zero-centered data is obtained that is the one used in the following steps.

Step 2: Calculate the covariance matrix

Covariance measures how much the dimensions vary from the mean with respect to each other. It is always measured between 2 dimensions in the following way 2.2.

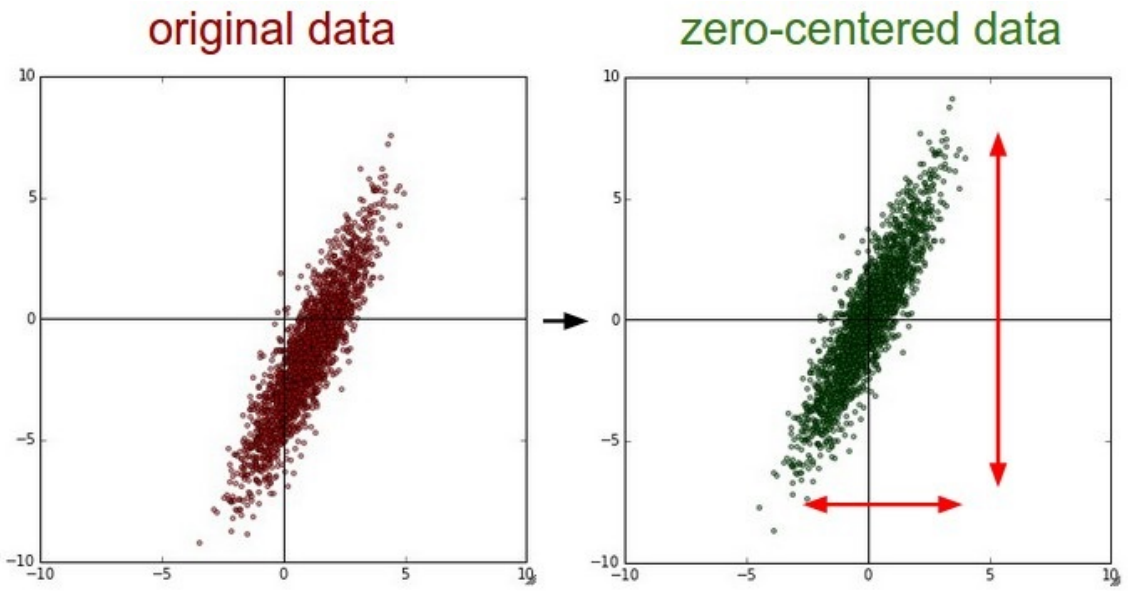


Figure 2.2: Left: Original 2-dimensional input data. Right: The data is zero-centered by subtracting the mean in each dimension. [2]

$$cov(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X}) \cdot (Y_i - \bar{Y})}{(n-1)} \tag{2.2}$$

If the covariance between one dimension and itself is calculated, the variance is obtained(eq 2.3).

$$cov(X,X) = \frac{\sum_{i=1}^{n}(X_i - \bar{X}) \cdot (X_i - \bar{X})}{(n-1)} = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{(n-1)} = var(X) \tag{2.3}$$

If the data set has more than 2 dimensions, there is more than one covariance measurement that can be calculated. For example, from a 3 dimensional data set (dimensions x , y , z) it could be calculated *cov(x,y)*, *cov(x,z)*, and *cov(y,z)*. In fact, for an n-dimensional data set, $\frac{n!}{(n-2)!\cdot 2}$ different covariance values can be calculated.

An useful way to get all the possible covariance values between all the different dimensions is to calculate them all and place them in a matrix.

$$C^{nxn} = (c_{i,j}, c_{i,j} = cov(Dim_i, Dim_j)) \tag{2.4}$$

For example in a dataset with three dimensions, the covariance matrix will be as described in 2.5.

$$C = \begin{pmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{pmatrix} \tag{2.5}$$

Step 3: Calculate the eigenvectors and eigenvalues of the covariance matrix

Since the covariance matrix is square, the eigenvectors and eigenvalues for this matrix can be calculated. The eigenvalues are rather important, as they show useful information about studied data.

By this process of taking the eigenvectors of the covariance matrix, the lines that characterise the data are extracted. The rest of the steps involve transforming the original data to be expressed in terms of these characteristics lines.

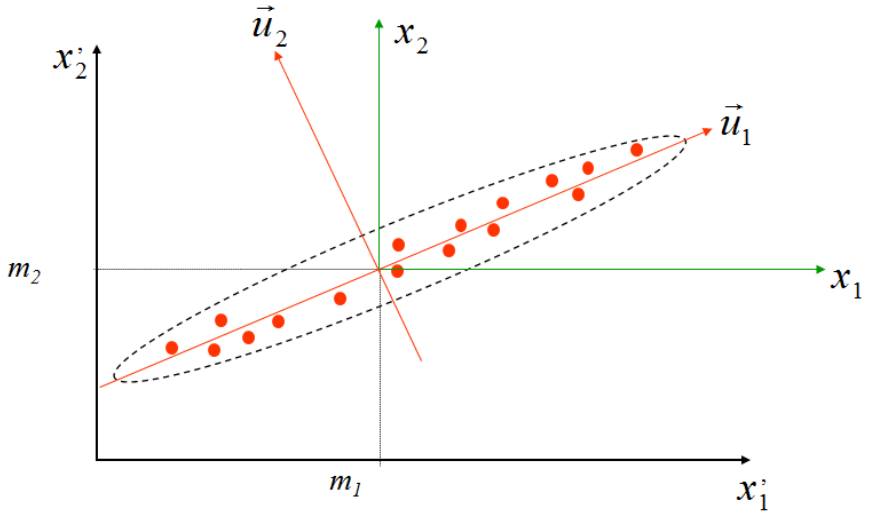


Figure 2.3: Eigenvectors u_1 and u_2 are obtained. It can be seen that u_1 is the line that characterise the data and u_2 does not offer big information [3]

Step 4: Choosing components and forming a feature vector

Once eigenvectors are found from the covariance matrix, the next step is to order them by eigenvalue, highest to lowest. This gives the components in order of significance.

At this moment, it can be decided to ignore the components of lesser significance. With this process some information is lost, but if the eigenvalues are small, there not exist much information loss.

Leaving out some components, the final data set has less dimensions than the original. To be precise, if originally the dataset has n dimensions, and therefore n eigenvectors and eigenvalues are calculated, and then only the first p eigenvectors are chosen, then the final data set has only p dimensions.

Finally, a feature vector is formed, which is just the name that is given to a matrix of vectors. This is constructed by taking the eigenvectors that are wanted to keep from the list of eigenvectors, and forming a matrix with these eigenvectors in the columns.

$$FeatureVector = \begin{pmatrix} eig_1 & eig_2 & eig_3 & \dots & eig_n \end{pmatrix} \tag{2.6}$$

Step 5: Deriving the new data set

This is the final step in PCA, once the components (eigenvectors) that have been chosen to keep in the data and a feature vector has been formed, the transpose of the vector is taken and multiplied it on the left of the original data set, transposed.

$$FinalData = RowFeatureVector \times RowDataAdjust \tag{2.7}$$

2.3.3. Singular Spectrum Analysis

In the field of time series analysis, a powerful technique has been developed which is known as SSA. This new technique can be employed in many practical problems such as the study of classical time series, multivariate statistics, multivariate geometry, dynamical systems and signal processing.

SSA can be applied in different areas as mathematics, physics, economics or financial. It can also be employed in other areas such as meteorology, oceanology, social science, market research and medicine.

Generally, any time series could be considered as an example of an application of SSA [28]

The basic SSA method consists of two complementary stages: decomposition and reconstruction. At the first stage the series are decomposed and at the second stage the original series are reconstructed and those reconstructed series are employed for forecasting new data points. The main concept in studying the properties of SSA is to know how well different components can be separated from each other.

SSA is described as a non-parametric technique that works with arbitrary statistical processes, independently of if they are or not linear, stationary or Gaussian. Therefore, in opposition to the traditional methods of time series forecasting, SSA method is non-parametric and makes no prior assumptions about the data. Furthermore, SSA method decomposes a series into its component parts, and reconstruct the series by neglecting the random (noise) component [29].

Considering a time series $X_T = (x_1, \ldots, x_T)$. Fix the window length L which needs to be smaller than the half of the total length ($L < T/2$) and K is defined as $K = T - L + 1$.

Step 1: Trajectory Matrix This step converts the one-dimensional time series $X_T = (x_1, \ldots, x_T)$ into the multi-dimensional series $Y_1, \ldots, Y_k$ with vectors $Y_i = (x_i, \ldots, x_{i+L-1})$. From this step the trajectory matrix $Y = [y_1, \ldots, y_K]$:

$$Y = (y_{ij})_{i,j=1}^{L,K} = \begin{pmatrix} x_1 & x_2 & x_3 & \ldots & x_K \\ x_2 & x_3 & x_4 & \ldots & x_{K+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_L & x_{L+1} & x_{L+2} & \ldots & x_T \end{pmatrix} \tag{2.8}$$

The trayectory matrix Y has all the elements along the diagonal $i + j$ have the same constant value which indicates that this matrix is a Hankel matrix.

Step 2: Singular Vector Decomposition (SVD) of YY^T At this step the eigenvalues and eigenvectors of the matrix YY^T and the represent it as $YY^T = PDP^T$. Where D is the diagonal matrix which contains the eigenvalues of YY^T ordered in values and P is the corresponding orthogonal matrix of eigenvectors of YY^T.

$$D = diag(\lambda_1, \ldots, \lambda_L) \tag{2.9}$$

where $\lambda_1 > \lambda_2 > \ldots > \lambda_L$.

$$P = (P_1, P_2, \ldots, P_L) \tag{2.10}$$

Step 3: Selection of eigenvectors: A group of l eigenvectors ($1 \leq l \leq L$) where each eigenvector are written as $P_{i1}, P_{i2}, \ldots, P_{il}$. The elementary matrix X_i is divided in different groups and the matrices are summed within each group. In this context, the set of eigenvectors indices are described as $I = i_1, \ldots, i_l$. The matrix X_i corresponding to the group I is defined as $X_I = X_{i1} + \ldots + X_{il}$.

Step 4: Reconstruction of the one-dimensional series: Finally, a new matrix with each term following $\tilde{X} = x_{i,j} = \sum_{k=1}^{l} P_{ik} P_{ik}^T X$ as an approximation to X. Now, averaging over the diagonals of the matrix $\tilde{X}$ the transition to one-dimensional series can be achieved.

2.3.4. Measurement Errors

A measurement error is defined as real value at the output of a measurement system minus the ideal value at the input of a measurement system:

$$\Delta x = x_r - x_i \tag{2.11}$$

Measurement errors can be divided in two main groups [30].

- Systematic error (bias) is a permanent deflection in the same direction from the true value and it can be corrected. Bias and long-term variability are controlled by monitoring measurements against a check standard over time.
- Random error is a short-term scattering of values around a mean value. It cannot be corrected on an individual measurement basis. Random errors are expressed by statistical methods.

In this work, only the systematic errors are taken into account and more concretely in two of them which are gain and offset errors. In this way, the random errors are depreciated for the difficulties to detect them.

In this way, the two measurement errors that this work focus on,namely offset and gain errors, are described:

The offset error, as shown in figure 2.4, is defined as the difference between the nominal and actual offset points and for an ADC the offset point is the midstep value when the digital output is zero. The offset error in percentage is compared against the full scale and this error affects all codes by the same amount.

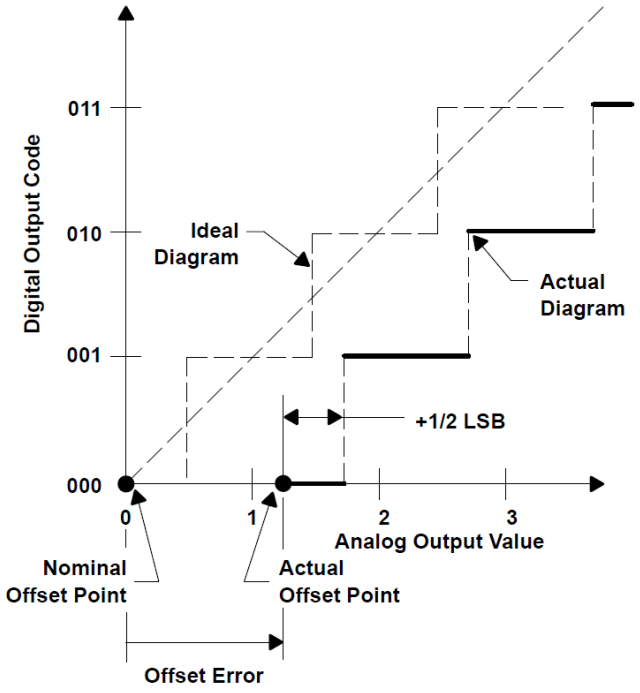


Figure 2.4: Offset error [4]

The gain error, as shown in figure 2.5, is defined as the difference between the nominal and actual gain points on the transfer function after the offset error has been corrected to zero. The gain point is the midstep value when the digital output is full scale.

In the example shown in figure 2.6 it can be seen how the function changes with the different errors. In this case, offset and gain errors are applied at the same time to a linear function. It can be noticed

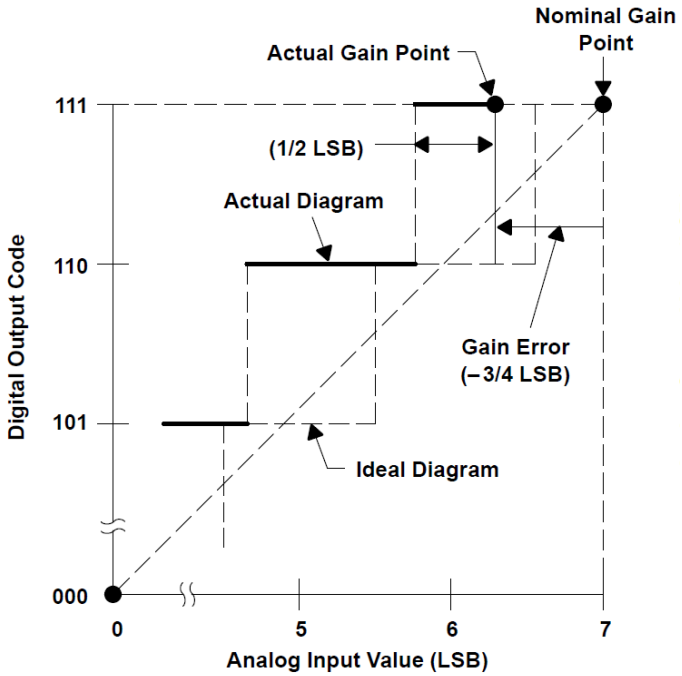


Figure 2.5: Gain errors [4]

how applying an offset to the function, the function moves through the ordinates axis. However, applying the gain error, the slope of the linear function changes.

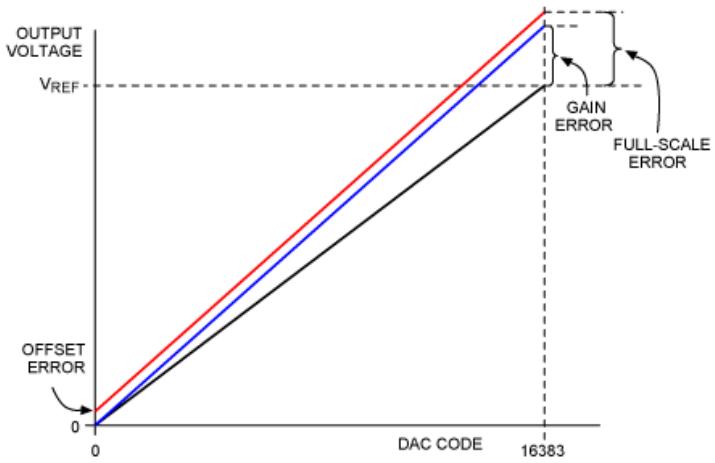


Figure 2.6: Gain and Offset errors [5]

These errors, in this work affect to the load measurements and this curve is not as simple as a linear function and how gain and offset errors affect to this curve is shown in chapter 3.

2.4. Conclusions

In this chapter, the employed tools have been explained. An introduction to each one of the employed tools has been given and in the bibliography some documents were provided where more detailed information can be found.

As explained before three tools are used, PCA for compressing the data, ANN is utilised for obtaining the load forecasting, and finally employing the theoretical principles of measurement errors it is possible to develop a system that is able to satisfy our targets as it is described in chapter 3.

Chapter 3

Work development

3.1. Introduction

In this chapter the complete system that have been used for satisfying the different goals is explained. Here, the tools and principles that were explained in the previous chapter are put together for developing a system that is able to suit the targets.

Furthermore, some details of this concrete system are given and the choice of some important parameters is analysed.

3.2. Overview of the system

In this section a general view on the employed system is discussed. Figure 3.1 shows an overview of the system. The available information pertains to the historical measurements from different SSs. The different electrical variables measured are: voltage, current, active and reactive power. However, for the system considered in this paper, just the current historical measurements are used. This is due to the fact that the equipment for current measurement has a big variability in the measurements what hinders the measurement error detection.

Therefore, with the available information, the developed system needs to be able to detect the different levels of gain and offset measurement errors. This system is divided into two steps for reaching the complete goal:

- In the first step, the available data is used for forecasting which the load consumption will be in the next day (d) in an hourly basis (24 load are forecast per day).
- In the second step, once that the real load measurements for the studied day (d) are available, they are compared with the forecast ones and the level of error is estimated by means of the developed error detection system.

In the first step, the data is prepared for be provided as input to an ANN. This process is explained in subsection 3.3.2. These inputs are compressed by means of PCA which is explained in subsection 3.3.1 and therefore the neural network training requires less time.

In the second and last step, the output obtained from the ANN, and the real measurements that have been obtained are compared by the method explained in section 3.5.

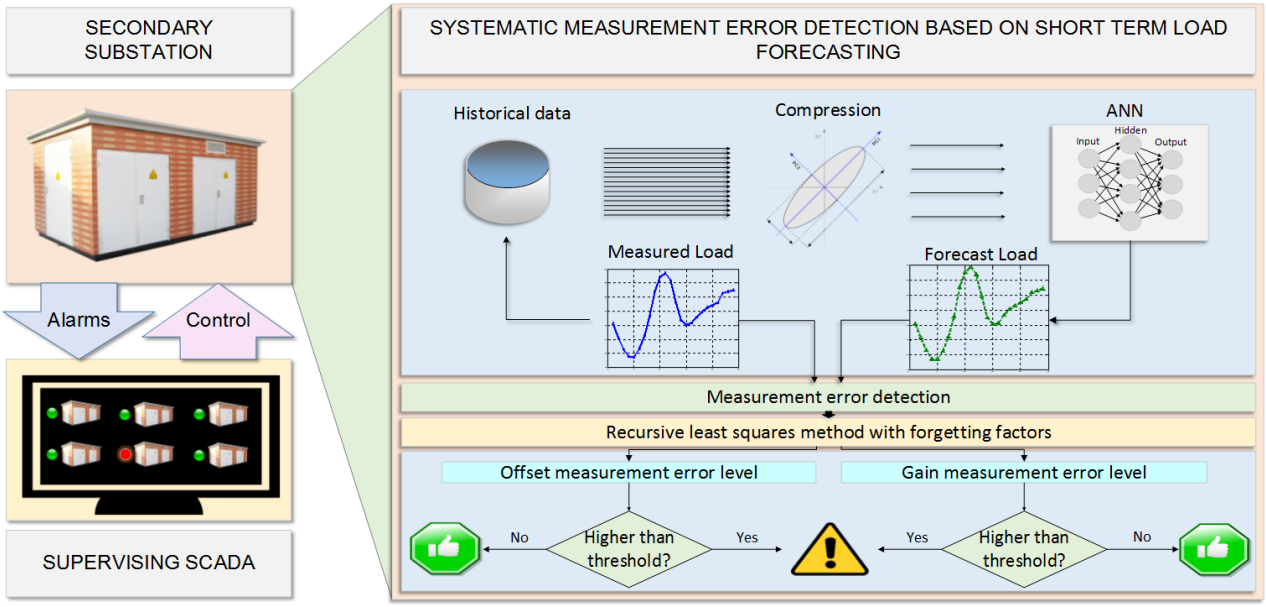


Figure 3.1: Overview of the general system

3.3. Load forecasting

In this section, the method for the load forecasting is explained. The only data that is available is composed by the historical measurements of a set of secondary substations. Following one of the method employed in [16], the system is able to perform an acceptable forecasting, using only the historical information. The chosen data for estimating what the hourly load in the day (d) will be is the following one:

- 24 hourly load of the previous available day (d-1)
- 24 hourly load of the previous available day (w-1)(d)
- 24 hourly load of the previous available day (w-2)(d)
- Day of the week (sine and cosine)
- Month of the year (sine and cosine)

Therefore, a total of 72 samples corresponding to hourly loads are employed. As the 72 first samples are very correlated between themselves, PCA technique can be employed to reduce the number of dimensions. The principal components resulting from the PCA process are now provided to the ANN and 4 more inputs corresponding to day of the week and month of the year is provided as well.

$$Input1 = sin\left(\frac{day}{7} \cdot 2 \cdot \pi\right) \tag{3.1}$$

$$Input2 = cos\left(\frac{day}{7} \cdot 2 \cdot \pi\right) \tag{3.2}$$

$$Input3 = sin\left(\frac{month}{12} \cdot 2 \cdot \pi\right) \tag{3.3}$$

$$Input4 = cos\left(\frac{month}{12} \cdot 2 \cdot \pi\right) \tag{3.4}$$

The day of the week and the month of the year are provided in sine and cosine ways due to the fact that in this way the range of values is between -1 and 1 and the difference between the last value and the first one is not high.

Therefore, the input vector that is provided to the ANN is composed by 4 inputs plus the number of principal components returned from the PCA process. A sketch of the system is shown in figure 3.2.

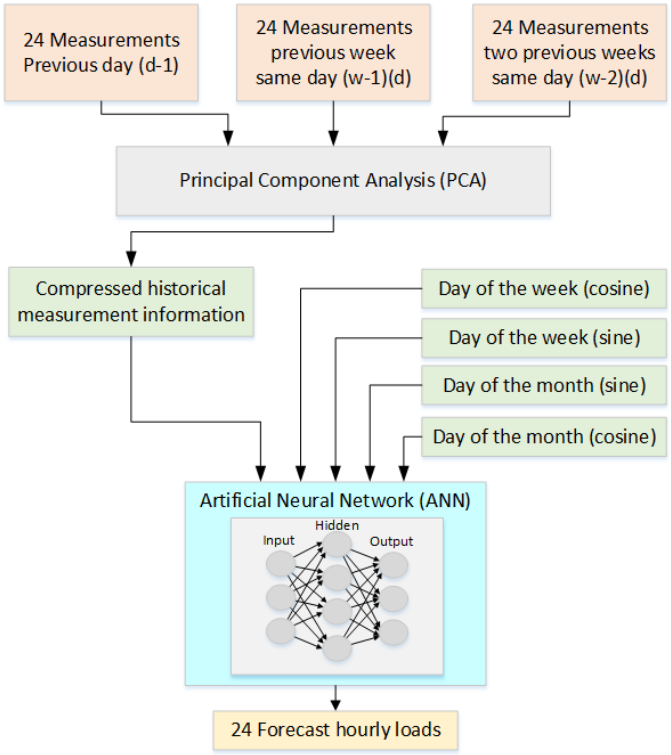


Figure 3.2: Description of the Load Forecasting system

3.3.1. Principal Component Analysis

As it was explained before, a set of 72 dimension data give historical information for forecasting. In this way, if this data set is directly provided to the ANN, it slows down the neural network training. As the 72 dimension shows the hourly load of different previous days, the correlation between the different dimensions are studied. As it is shown in table 3.1 there is a big correlation between the hourly data.

Hour	01:00	02:00	03:00	04:00	05:00	06:00	07:00	08:00	09:00	10:00	11:00	12:00
01:00	1,00	0,92	0,85	0,81	0,77	0,80	0,72	0,57	0,57	0,68	0,72	0,68
02:00	0,92	1,00	0,95	0,91	0,88	0,88	0,79	0,63	0,57	0,67	0,77	0,78
03:00	0,85	0,95	1,00	0,96	0,93	0,93	0,86	0,70	0,61	0,69	0,80	0,82
04:00	0,81	0,91	0,96	1,00	0,97	0,95	0,90	0,76	0,67	0,73	0,83	0,84
05:00	0,77	0,88	0,93	0,97	1,00	0,96	0,93	0,81	0,73	0,76	0,84	0,86
06:00	0,80	0,88	0,93	0,95	0,96	1,00	0,94	0,81	0,73	0,77	0,82	0,84
07:00	0,72	0,79	0,86	0,90	0,93	0,94	1,00	0,93	0,85	0,84	0,84	0,85
08:00	0,57	0,63	0,70	0,76	0,81	0,81	0,93	1,00	0,95	0,87	0,78	0,78
09:00	0,57	0,57	0,61	0,67	0,73	0,73	0,85	0,95	1,00	0,93	0,78	0,74
10:00	0,68	0,67	0,69	0,73	0,76	0,77	0,84	0,87	0,93	1,00	0,89	0,82
11:00	0,72	0,77	0,80	0,83	0,84	0,82	0,84	0,78	0,78	0,89	1,00	0,94
12:00	0,68	0,78	0,82	0,84	0,86	0,84	0,85	0,78	0,74	0,82	0,94	1,00

Table 3.1: Covariance table between the first 12 hours of a day using data from a real secondary substation. A total of 605 days were studied.

It is a good scenario to apply PCA and reducing the data dimensionality. Utilizing the PCA MATLAB toolbox, and providing the previous available information (600 days approximately) the principal component are obtained.

The percentage of the total variance for each of the 72 components is returned ordered by the percentage of variance as it is shown at table 3.2. In this table, it can be noticed that the first PCs represent the information the most. However, the last PCs basically do not store information and therefore it can be dispensable.

PC 1-12	PC 13-24	PC 25-36	PC 37-48	PC 49-60	PC 61-72
79,1659805	0,34157775	0,08589552	0,04661787	0,02618346	0,01320213
4,84705933	0,31395501	0,07665263	0,04574963	0,02483364	0,01265725
3,32608959	0,26653706	0,07260811	0,04367215	0,02378294	0,01198498
1,96242628	0,22365902	0,07161773	0,04041543	0,02279032	0,01122793
1,80536065	0,21653948	0,06976929	0,03884886	0,02239577	0,00986127
1,67388064	0,19336858	0,06536597	0,03622134	0,02057482	0,00958681
0,89809079	0,16982107	0,06310965	0,03521733	0,01976735	0,00675311
0,62905314	0,13307761	0,06097508	0,0329981	0,01906753	0,00660815
0,5643196	0,12855429	0,05772813	0,03116253	0,01790833	0,00572152
0,47962091	0,11613771	0,05729455	0,02998113	0,01643708	0,00488968
0,39390509	0,1081835	0,05507329	0,02918008	0,01611361	0,00438262
0,37754167	0,09333693	0,05269626	0,02783014	0,01447645	0,0040363

Table 3.2: Principal Components percentage of variance: In this table the percentage of variance of each of the 72 Principal Components are represented.

The more PCs are not considered, the faster the neural network training is, but more amount of information is lost. Taking these points into account, it was decided to keep the enough number of PCs to keep the 97% of the variance. As shown in the example, this would mean to keep the first 19 PCs. In this way and at this concrete example, the number of dimensions is reduced from 72 to 19 whit just a lost of a 3% of variance. In the different cases the number of dimensions is reduced from 7 dimensions

(simulator data) to 29 dimensions (real SSs data).

3.3.2. Artificial Neural Network

As described previously, the ANN receives the output PCs obtained from the PCA procedure and 4 more inputs divided in:

- Day of the week (sine and cosine)
- Month of the year (sine and cosine)

These inputs are stored in a matrix of $n \times m$ dimensions where n is the reduced number of inputs (11 to 33) and m is the number of samples (600 to 800 depending on the number of employed days). A target matrix is as well created with $24 \times m$ dimensions which stores the real hourly load values measured in the day d.

Each matrix is divided into two sub-matrix. The value p represents the portion that is stored in the first sub-matrix (0.8-0.95). The first sub-matrix is an $n \times (m \cdot p)$ matrix and the second sub-matrix is an $n \times (m \cdot (1-p))$.

The first sub-matrix is used for the neural network training which also is divided in three groups for training, test and validation of the ANN. With the second sub-matrix the ANN faces new data that has not seen before. In this way, a case close to the real one is performed, where previous data is used for training and then new data is received as days pass by.

At this time, providing to the ANN the first sub-matrix corresponding to input and target matrices, the ANN is trained. This data is divided by the neural network Matlab toolbox in three groups. Firstly, a group corresponding the 60% of the data for the real training, 20% for the testing and the final 20% for the validation. Indicating the number of neurons and the algorithm, the train assigns the corresponding weight to each neural connection for reaching the closest to the target with the given inputs.

The ANN used in this paper is a three-layer perceptron with a single hidden layer since it is the most frequently employed in forecasting and time series due to its performance [31]. The output and the input signals are related by the following equation:

$$y = \varphi\left(\sum_{j=1}^{n} x_j \cdot w_j - \theta\right) \tag{3.5}$$

where y is the output, x_j represents the input data, w_j expresses the weight that is associated with each x_j, and θ is the threshold. Finally, φ is the transfer function that usually is represented as:

$$\varphi = \frac{1}{1+e^{-x}} \tag{3.6}$$

The Levenberg-Maquardt algorithm has been chosen for the ANN training two reasons:(i) the number of iterations required for the ANN training is lower in comparison to other techniques,(ii) this algorithm always guarantees the convergence of the training process[22][23].

Taking into consideration that every EIT to be checked by the described system generates its own data, the ANN differs between the different EITs, hence the importance of selecting the algorithm that best provides accuracy and fast convergence in equal measure.

An important issue is to define the number of neurons of the hidden layer since this significantly affects the performance of the network in terms of execution time and accuracy. In this regard, an iterative process which consists in carrying out different tests is used. Varying the number of neurons of the hidden layer from 1 to 100 and comparing the performance for each iteration, the ANN that minimizes the test error is composed of a 15-neuron hidden layer.

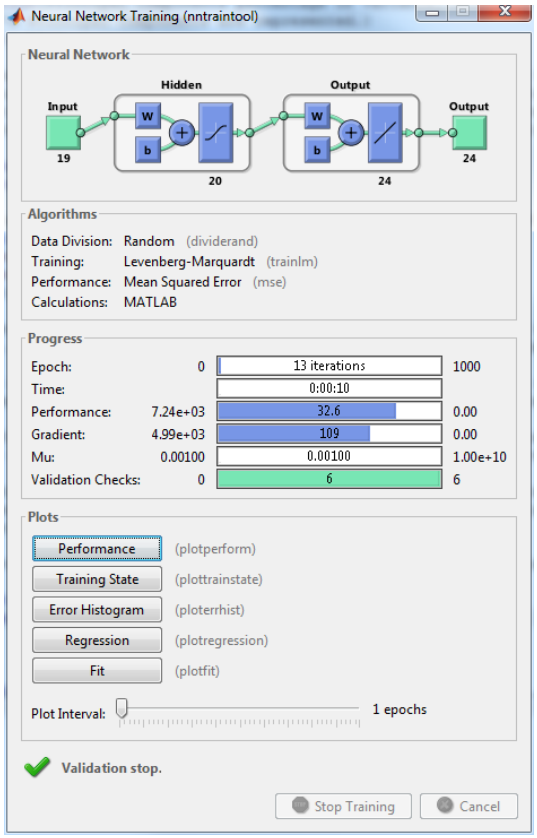


Figure 3.3: MATLAB toolbox GUI for neural training

For adjusting the number of neurons, after several tests where the error was measured using different number of neurons for the same data, it was shown that the best result was given with a hidden layer of 5-20 neurons as it can be seen in figure 3.4. This error is calculated using the second sub-matrix, the forecasting for the day d, is estimated using the previous information and the *Mean square error (MSE)* is calculated comparing the forecast load against the real load in day(d).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2 \tag{3.7}$$

where n stands for the total number of samples, Y represents the actual load value and finally $\hat{Y}$ corresponds to the forecast value.

3.4. Error Measurement Detection

3.4.1. Errors identification

Once that the load forecasting has been carried out and the new measurements for the day (d) are available, the error measurement detection process can start. Firstly, the different errors that can be

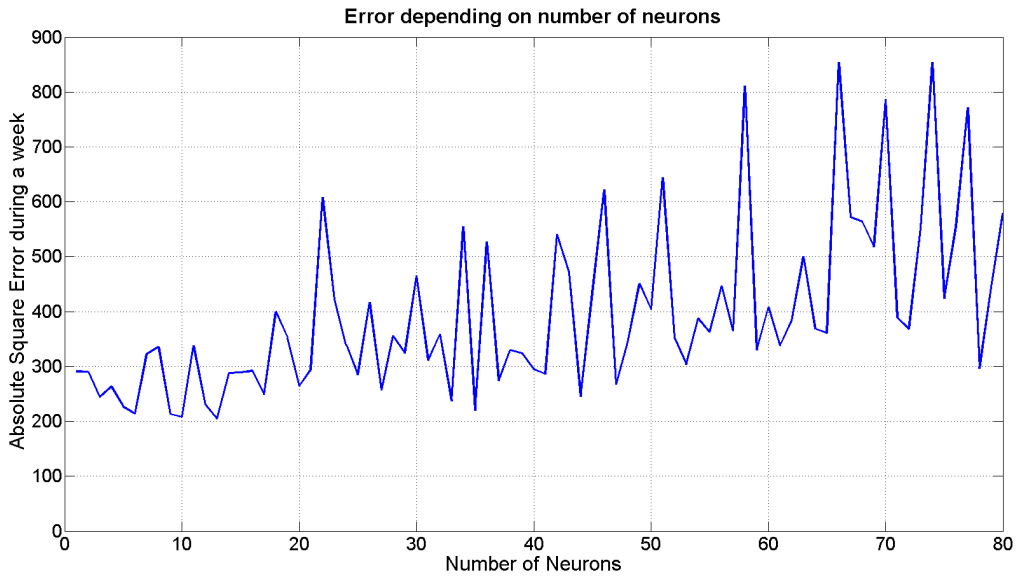


Figure 3.4: Mean absolute square error depending on number of neurons

detected with this system are described. These errors are systematic ones and they are classified in two different types:

- Offset error
- Gain error

3.4.1.1. Offset Error

As it was explained in section 2.3.4, an offset error affects all codes by the same amount and focusing in the secondary substation measurements, an offset error would affect as it is shown in figure 3.5.

In this way, the load measurement that is received (Measured load) could be represented as the load measurement with no error (Ideal load), plus a constant which represents the offset error (β):

$$MeasuredLoad = IdealLoad + \beta \tag{3.8}$$

3.4.1.2. Gain Error

As it was explained in section 2.3.4, a gain error affects depending of the measured value and focusing in the secondary substation measurements, a gain error would affect as it is shown in figure 3.6.

In this way, an offset error could be represented as:

$$MeasuredLoad = IdealLoad \cdot \alpha \tag{3.9}$$

3.4.1.3. Error combination

In some cases, both errors could appear at the same time which makes more difficult to identify the level of each one of the errors. In that case the Real load is represented as:

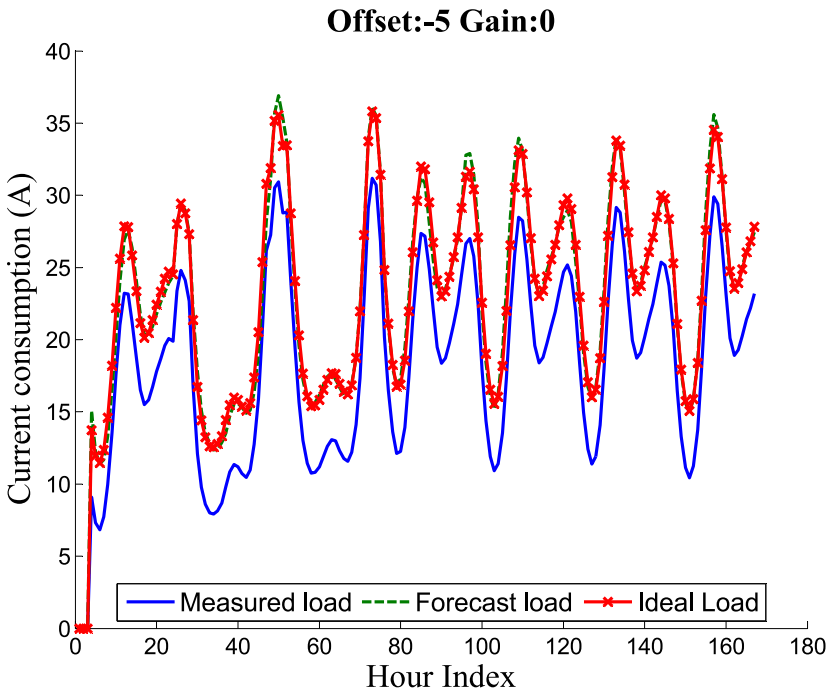


Figure 3.5: Offset error -5%

$$MeasuredLoad = (IdealLoad \cdot \alpha) + \beta \tag{3.10}$$

3.4.2. Errors detection

Regarding their presence, offset and gain errors can occur simultaneously, which greatly hinders the error detection. In this context, both errors must be detected independently. Interestingly, the system described in this paper performs well under worst-case scenarios such as the occurrence of both errors with different signs.

The most general possible scenario is considered when both errors appear at the same time as described in the following equation:

$$m_{lp}(t) = (i_{lp}(t) \cdot \alpha) + \beta \tag{3.11}$$

where m_{lp} represents the measured load profile, i_{lp} is the ideal measurement load profile, i.e. when no error occur, and α and β represents the gain and offset error, respectively.

By using the strategy described previously, the forecast load profile ($f_{lp}(t)$) can be estimated. Since the process of forecasting introduces a random error ($e_f(t)$), equation 3.11 can be rewritten as:

$$m_{lp}(t) = ((f_{lp}(t) + e_f(t)) \cdot \alpha) + \beta \tag{3.12}$$

$$= f_{lp}(t) \cdot \alpha + e_f(t) \cdot \alpha + \beta \tag{3.13}$$

It goes without saying that the better the load prediction, the smaller the error introduced. However, this error can be neglected on account of the fact that it is a random error with mean 0. Therefore, by using different averaging processes, equation 3.13 can be reformulated as:

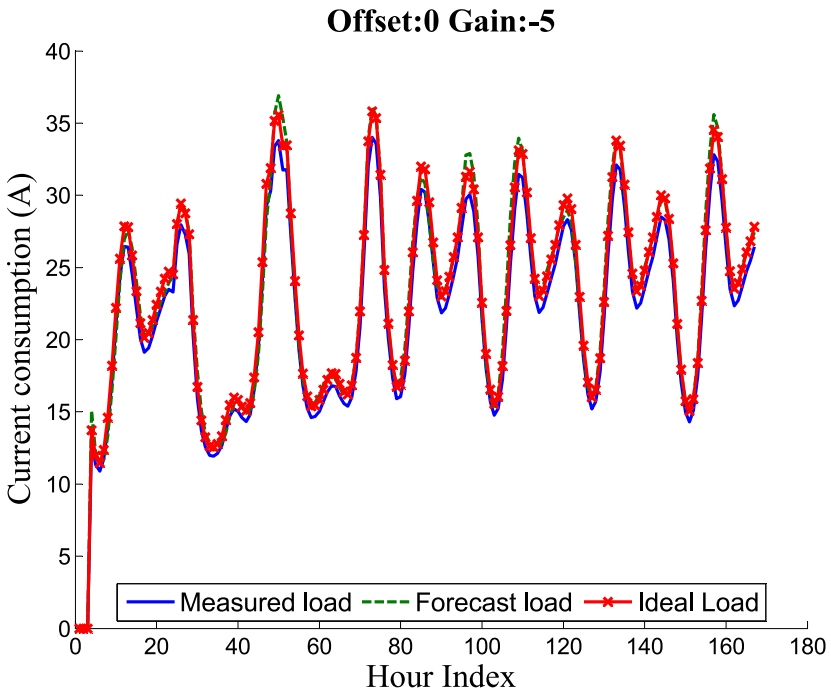


Figure 3.6: Gain error -5%

$$\overline{m_{lp}}(t) = \overline{f_{lp}}(t) \cdot \alpha + \beta \tag{3.14}$$

Differentiating both terms in equation 3.14 leads to equation 3.15:

$$\frac{\Delta\overline{m_{lp}}(t)}{\Delta t} = \frac{\Delta(\overline{f_{lp}}(t) \cdot \alpha + \beta)}{\Delta t} = \frac{\Delta\overline{f_{lp}}(t)}{\Delta t} \cdot \alpha \tag{3.15}$$

Then, the following equation can be obtained by solving for α:

$$\alpha = \frac{\Delta\overline{m}_{lp}(t)/\Delta t}{\Delta\overline{f}_{lp}(t)/\Delta t} = \frac{\Delta\overline{m}_{lp}(t)}{\Delta\overline{f}_{lp}(t)} \tag{3.16}$$

Finally, substituting the gain error into equation 3.14, and solving for β leads to 3.17:

$$\beta = \overline{m_{lp}}(t) - \overline{f_{lp}}(t) \cdot \alpha \tag{3.17}$$

The proposed system is able to estimate the value of gain and offset error independently. As there exist a considerable variability in the offset and gain measurement error estimation, the recursive least squares algorithm with forgetting factors (FRLS) is employed for filtering. This algorithm has been extensively utilized in the time-varying system as in [32] where a novel recursive least squares with forgetting factors algorithm was developed.

With this process, the system is able to estimate the value of gain and offset error independently and therefore if an error remains during a period of time, out of a defined threshold range, it alerts of a misbehaviour of the measurement device.

3.4.3. Real system description

In this subsection, the details for the real employed system described before are given. Firstly, it is supposed that the data provided does not contain any offset or gain error because that data is employed for training the ANN and then the errors are injected manually over the data. In the real case, this data is used for the network training, and the new one would be analysed for detecting the error levels.

The complete system was developed in MATLAB R2014a and the standard functions available in Matlab libraries were employed for file reading, data processing, principal component analysis, artificial neural network and process system.

In this work, as the information available is supposed to not contain any error, the errors are included manually as described in equation 3.10. In this way, the "Ideal Data", is transformed in a "Real Data" with errors and this is the data that the system receives.

As the data and the forecast data could present some variability in the measurement or experimental error samples (outliers), which would be very dangerous for the differentiation, a first **averaging between 5 consecutive samples** is done. This average could be considered as a digital filter which well employed improves greatly the forecasting results.

As it was described in the previous section, for the process, it is needed to differentiate the "Real Load" and the forecast one. In this way, the differentiation is applied over an interval of time. After different tests, the interval of time which was selected is **7 days**. This differentiation is highly important for the gain level error successfully detection.

With the goals of reducing the forecasting error contribution to this system and smooth the derivation curves, a second **averaging of 2 samples** is done in both, differentiated forecasting curve and differentiated real load curve.

Finally, division is done for obtaining the gain value, for each of the n different samples of the two differentiated curves and therefore n values of gains are obtained. The values that are **higher than 200% (2) or smaller than 50% (0.5)**, are depreciated. With the other ones, the gain estimated value is the **median of all these values**.

This process is done with every one of the studied hours, and therefore a different result of gain error can be estimated for each hour. Finally, for avoiding false alarms a filtering process is carried out. As mentioned before this filtering is carried out by a recursive least squares algorithm with forgetting factors (FRLS) where **the forgetting factor utilised was 0.83**. This factor was decided by comparing the obtained filtered curves at different forgetting factor. This factor was chosen because is considered the one that gives a good relationship between filtering false alarms and fast detection of the measurement error.

For making it easier, some of the parameters explained above can be modified by means of the developed GUI which is explained in appendix A.

3.5. Forecasting optimization with SSA and ANN combination

As it has been described in section 3.3 the more accuracy the forecast process has, the better results in erroneous measurements detection is obtained. In this way, a hybrid system between SSA and ANN is also developed.

Hybrid methods between SSA and ANN can be a good combination in terms of load forecasting. In this work, two different topologies have been implemented.

3.5.1. Hybrid Model 1: SSA filtering

In this model, SSA has be employed for data decomposition and then select exclusively the components that have a considerable amount of information and they are employed for reconstruct the filtered original information in a time series array. Concretely, a total of 50 principal components are employed to reconstruct the original data.

In this way the noise is eliminated and the information that arrives to the ANN, which conserves the structure explained in 3.3.2, is cleaner which can improve the forecasting result in terms of forecasting errors.

The output obtained from the SSA process is a time series array which needs to be restructured for providing the correct input to the ANN. Following the structure of 3.3.2 it is needed a total of 76 dimensions. Unlike PCA where the set of information was compressed reducing the number of dimensions and that was the input to the ANN, here SSA does not compress the information and the 76 dimensions are employed as input.

3.5.2. Hybrid Model 2: Principal Component separated ANNs

In this second method, the time series information is decomposed by the SSA method. The information to decompose is at a first step analyzed to detect which principal components contain valuable information and which ones contain noise. In this second method, each of the principal components that contain valuable information, has its own trained ANN.

This personalized training for each ANN associated to each principal component, follows the same data structure as the previous models. It means that the data contained in every principal component is restructured for creating a 76 dimensional data with the information for the historical data (72 dimensions) and the day of the week and the month of the year. The main goal in this hybrid model is to forecast the value for each principal component in the required hour. the output of the different ANN gives the forecast values for all the valuable PCs which are reconstructed to get the forecast values that are employed in the measurement error detection.

3.5.3. Hybrid Model Combination

The two hybrid models described in previous section provide two forecast concurrently. It was noticed that both of forecast method could be combined and in this way one of the method can reduce the other method error. It would prevent the over or underestimation of any of the two models. For comparing both methods, the *Mean Absolute Percentage Error (MAPE)* indicator is employed which was especially designed for prediction accuracy and it is described by

$$MAPE = \frac{1}{n} \cdot \sum_{i=1}^{n} \left| \frac{A_i - F_i}{A_i} \right| \cdot 100 \ \ (\%) \tag{3.18}$$

where A_i stands for the actual value and F_i represents the forecast value. The MAPE for each of the previous hybrid model was analysed and it returned an averaged MAPE of 3.99 % for the Hybrid model 1 and a MAPE of 5.15 % for the Hybrid model 2. The first idea for the combined method is to employ the average between the two forecast data set, and the MAPE was reduced to 2.68 %. The value of 2.68 % results a great improvement against the two method separately. However, a different approach was developed where the average would be inversely weighted to the MAPE value related which each model.

By employing this weighted average, the MAPE value was slightly improved and it was reduced to 2.60 %.

In Fig. 3.7 the real measurements are compared with the forecast measurements obtained with hybrid models 1 and 2. It can be seen how the real measurement is found between the two forecast model. Representing as well the forecast measurements obtained through the hybrid model combination it can be realised of the forecast improvement which was reflected through the MAPE value.

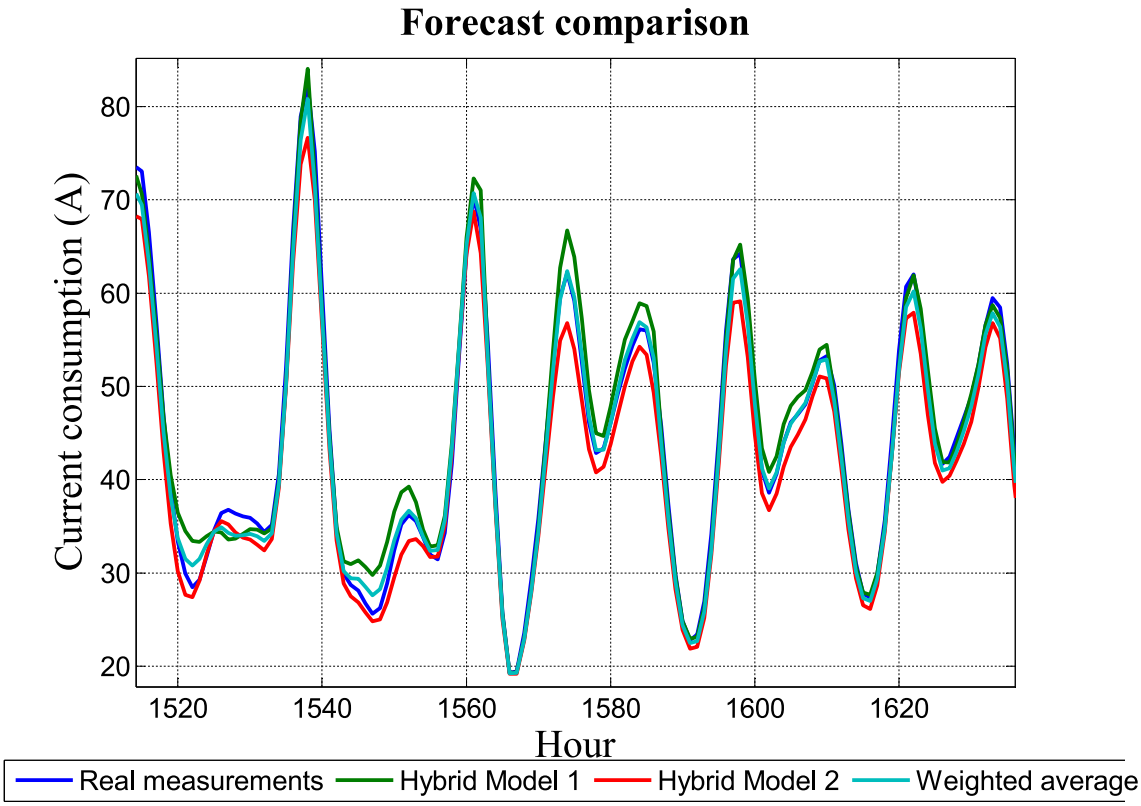


Figure 3.7: Hybrid models forecast comparison

The different MAPE values were collected and represented on a bar diagram in Fig. 3.8 which the weighted average hybrid model combination appears as the method with best accuracy.

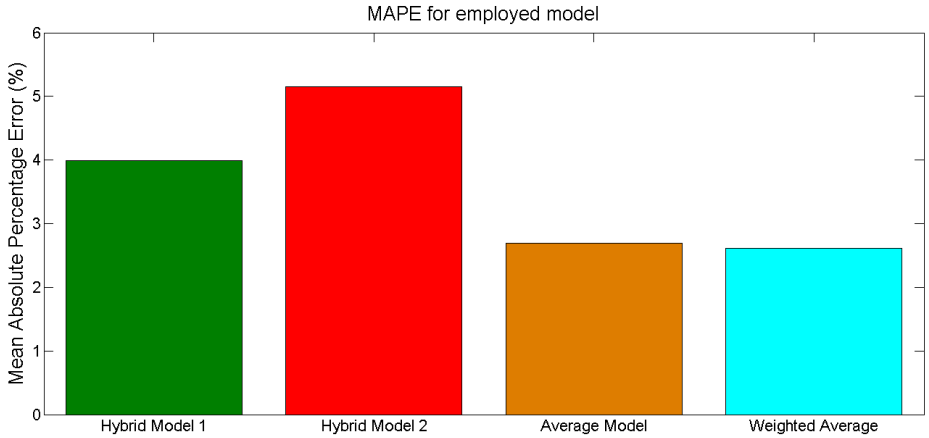


Figure 3.8: Hybrid models forecast MAPE comparison

However, as the load forecasting method described in subsection 3.3, presents a less data dimension-

ality which facilitates the computational time and the data storage, it will be the one employed hereafter.

3.6. Conclusions

In this section the complete process carried out for completing this work has been explained and the main details of the system have been given.

Starting with the load forecasting process and describing the main tools utilized. The PCA techniques that were used for information compression and then the artificial neural network that was developed using the information that the PCA returns, the estimation of which the future load will be is provided.

Once that the forecast load has been predicted and the real data for the corresponding day is available, they both are compared with each other and the different levels of gain and offset error are estimated. The comparison is made by differentiation and with different averaging processes.

Finally, a forecasting optimization by the combination of ANN and SSA has also been studied showing forecasting enhancement but incrementing the computational and storage costs.

In this way, basing this system in the theory described in chapter 2, it has been possible to detect the different errors as it will shown in next chapter 4.

Chapter 4

Results

4.1. Introduction

In this chapter, the developed system described in chapter 3 is tested against simulated and real data. In this way, it is possible to demonstrate if the developed system meets the targets that were proposed at the beginning of the book and if it is able to detect the gain and offset measurement errors.

Then, the validation of the results is done with different indicators such as MAPE, *Normalized Mean Bias Error (NMBE)* and *Normalized Root Mean Square Error (NRMSE)* and with the absolute error between the detected error and the real one. This verification is applied over both real and simulated data.

4.2. Data bases

For the testing of the system, two different data bases are employed, one with simulated data and the second one with real data.

As far as the first data set is concerned, the simulator has been developed by a distribution operator and it provides simulated measurements from different SSs located in the Henares corridor, in Madrid. This simulator provides 4 measurements: Voltage, Active Power, Reactive Power and Current. For this work just the current measurement is used which is the parameter that has been employed in the literature and because the equipment for current measurement has a big variability in the measurements what hinders the measurement error detection.

A simulated data set based on current measurement is therefore obtained every 15 minutes for two years (from January 1st, 2010, to December 31st, 2011.

Simulator provides the data in raw text, and it was needed to import that data, using different scripts to Matlab. Once that it was imported it was possible, by doing an averaging process between every 4 measurements, to obtain an averaged hourly load. In the simulated case the data is very correlated between the days which enhances our system.

On the other hand, the system has also been tested against a real data set obtained from different SSs located in the Community of Madrid. This data set contains information half-hourly recorded from several SSs in the period ranging from January 1st, 2010 to December 31st, 2011.

Although both data sets contain data acquired with different frequency, they are analysed in the same way since only hourly average values are taken into consideration.

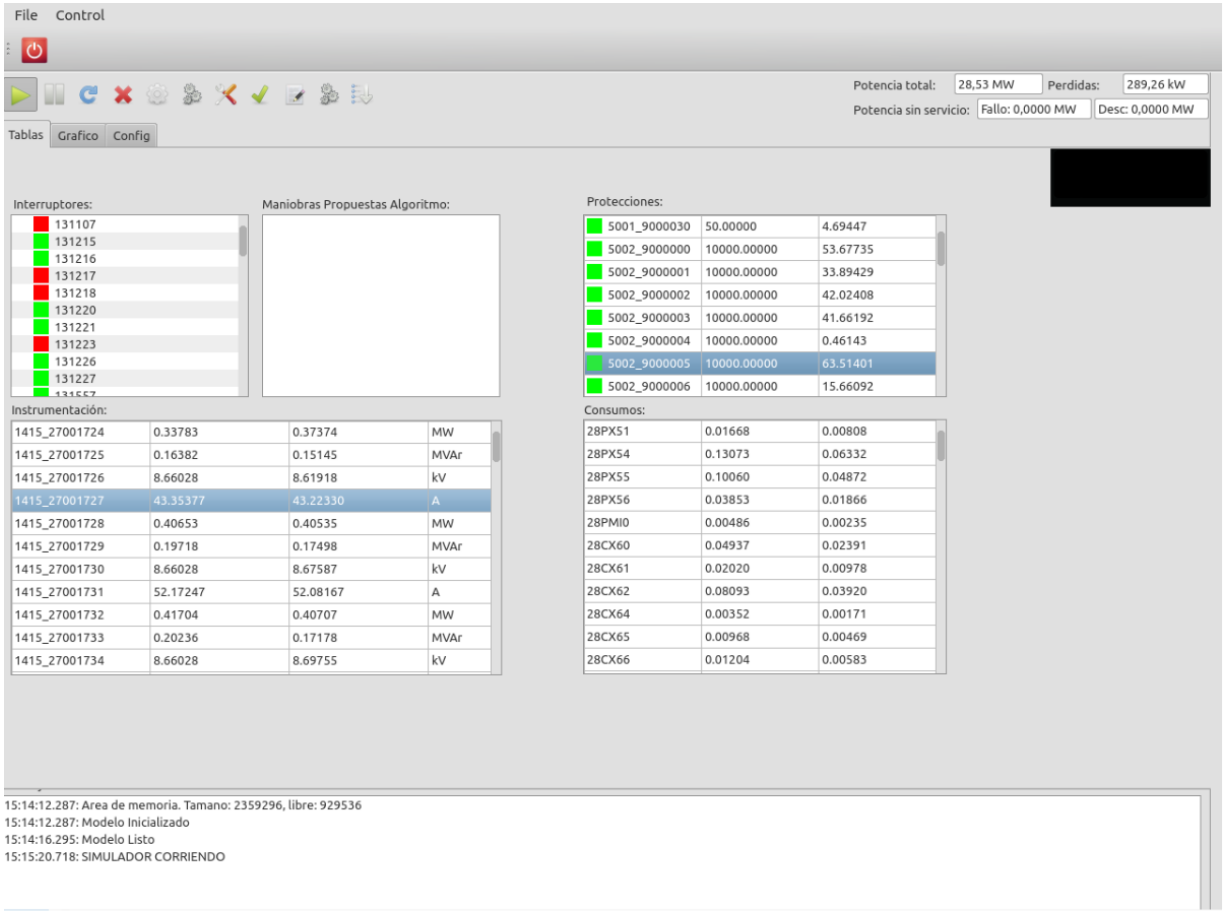


Figure 4.1: Distribution operator simulator screenshot

In the real data, some measurements are missing. That is very harmful to our system, due to the fact, that a missing measurement affects to the current day, to the day after, to the same day in the week after and to the same day two weeks ahead. For solving this, when this occurs and when it is possible, the measurement that was taken one year before is replaced.

4.3. Experimental Scenario

Both data bases are analysed in the same way, independently if they have 2 or 4 measures per hour, an averaging process is carried out for just obtaining one value per hour and applying the same process to both data.

Then, the data is divided in two subgroups, most data is employed for training (80-95%) and the rest is used for testing. In the data employed for training (80-95%), the neural network also divides the data in 3 subgroups for training, validate and testing the network. This division was set to 60 % for training, 20% for validation and 20% for testing. In his work, it was decided to make a previous division because,in this way, it is closer to the real case, where it is needed to train the neural network with the available information and once it has been trained, it faces new data of the following days for detecting the error level in those days.

Under normal operating conditions, the current measurement accuracy including all the equipment involved in the measuring process (mainly current transformer and measurement equipment) is below ±2%. The contribution of the current transformer is not higher than ±1% [33] [34] and the measurement

equipment [35] can include between ±0.5% and ±1% error depending on the manufacturer.

Within this context, for ensuring that a systematic measurement offset or gain error is occurring without generating false alarms, a minimum threshold of ±5% is considered. Nonetheless, the results have been analysed until a maximum measurement error of ±10%.

4.4. Experimental Results

In this section, the system is tested against both simulated and real data. It shows the result of the different steps of the system, the forecast measurements are depicted against the real ones and in this way it is possible to appreciate how is the forecasting system working.

Then, different levels of gain and offset errors between -5% and -10% or between +5% and +10% are applied and our system returns an estimated error value per hour. As it is known that the error remains in the same level, the estimated error levels are filtered by a recursive least squares with forgetting factor for ensuring than an error is occurring and not false alarm are appearing.

As it will be shown there exists a big difference between working with the real data or the simulated one and it can be perceived in the results. The real data is less correlated between the days, and in this way it provides difficulties to carry out a good load forecasting. If the system is not able to have an acceptable prediction, the forecasting error increases and in this way the process explained in the previous chapter where the forecasting error was depreciated for estimating the gain and offset error will not be completely valid and our targets will not be completely satisfied.

It is also important to highlight that as a forecasting error is always appearing and it is a random error, there is variability between the error detected in the different days. This variability is also higher in the real data case, due to the fact, that the forecasting error is higher as well. That is the reason why it was decided, as explained before, to approximate with the line that best fits these points.

4.4.1. Simulated data

In the following figures, the results obtained from testing the described system against simulated data are depicted. Firstly, in figure 4.2, all the measurements taken from 2 years (730 days) are shown.

Then, in figures 4.3, 4.4 and 4.5 it is shown a comparison between the forecast and the real measurements in different ways. Figures 4.3, 4.4 compare forecast and simulated measurements at the same hour during all available days that have not been used for training. Finally, figure 4.5 depicts the 24 hourly loads forecast for a day d and the available measurements for that day, in this case the error between the forecast and the simulated measurement data can be neglected.

As the prediction is good in the simulated case, the error detection will give a better result. In both cases, the output that shows the level of error has variation between the days although the included error is the same. That is the reason why the output of error level for all the different days will be approximated with the line that fits best. This line will determine the error level that the EIT is experiencing.

Finally, there are several figures (4.6, 4.7, 4.8, 4.9) where the results with different error levels (offset and gain errors applied simultaneously in some cases) are shown. As there is not a considerable forecast error in simulated data, the error detection will be acceptable.

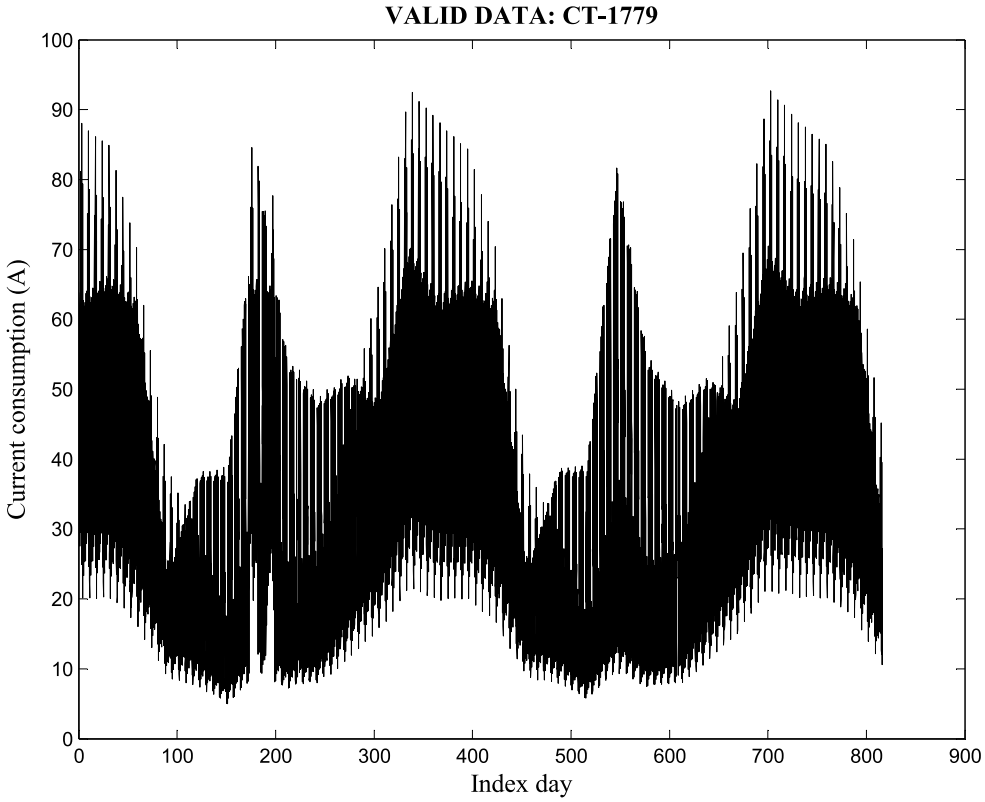


Figure 4.2: CT 1779 - approximately two years measurements: Measurements taken during 815 days are depicted in this figure.

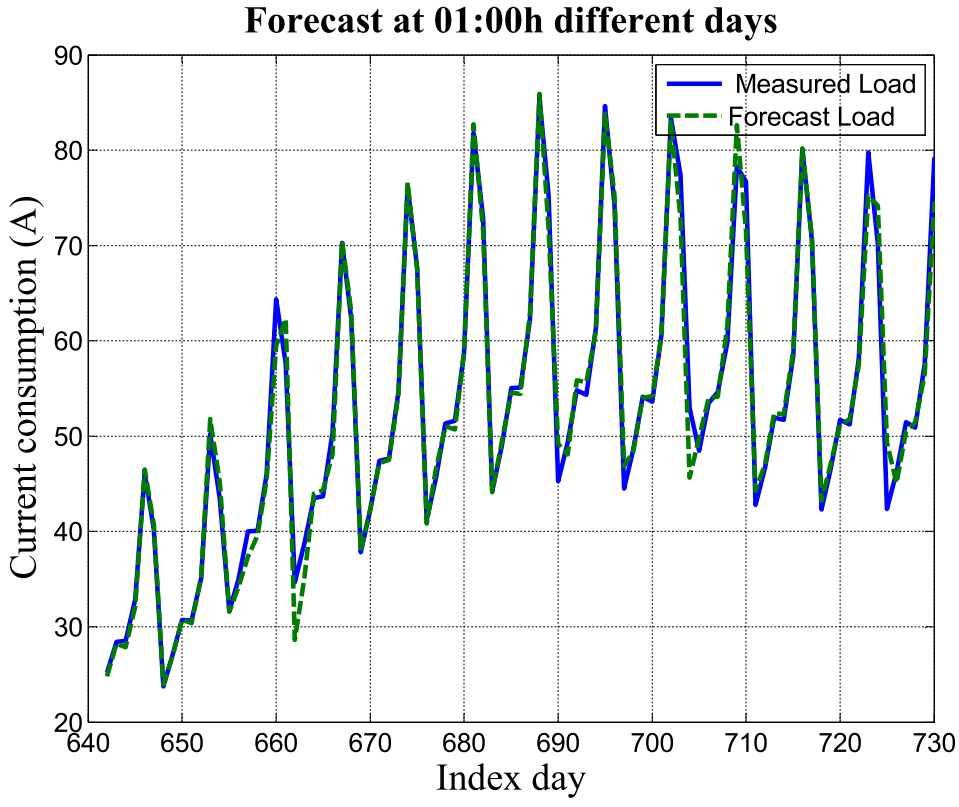


Figure 4.3: CT 1779 - Forecasting at 01.00h: During available days the measurement at 01:00h and the forecast for that hour are depicted.

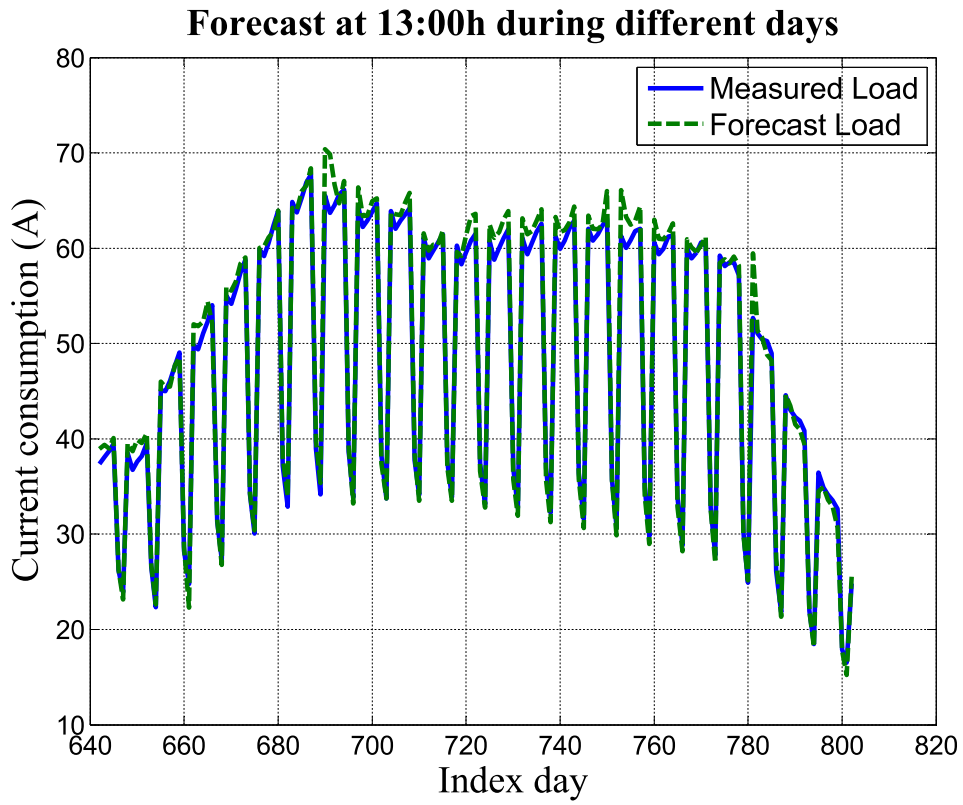


Figure 4.4: CT 1779 - Forecasting at 13.00h: During available days the measurement at 13:00h and the forecast for that hour are depicted.

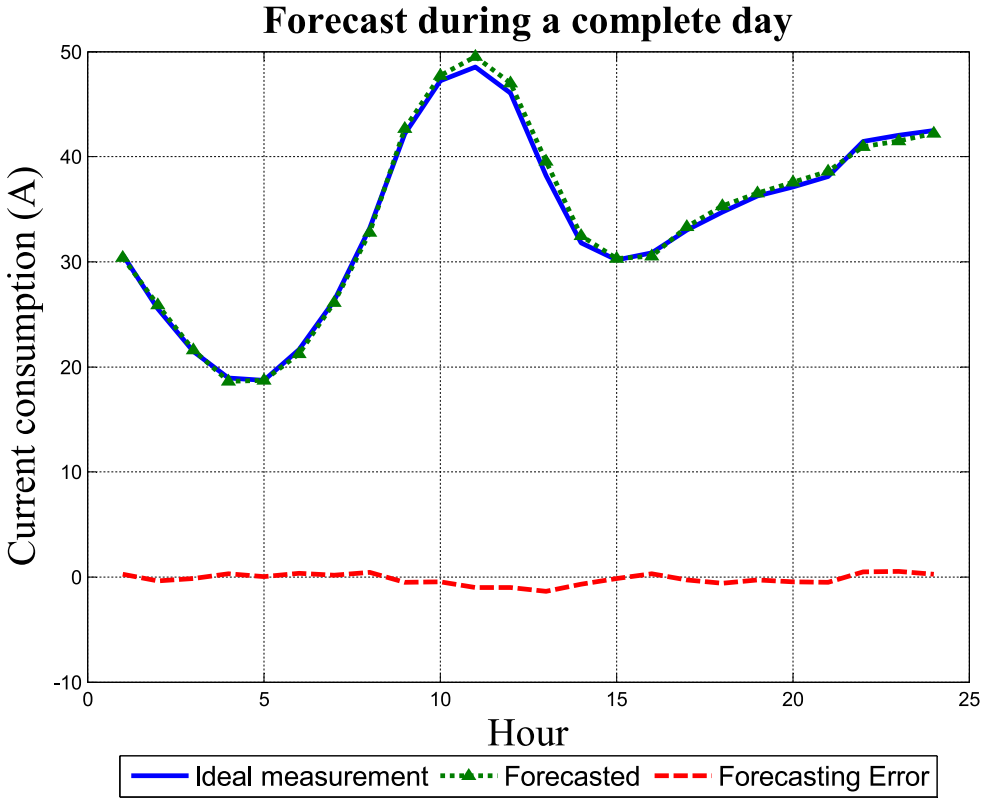
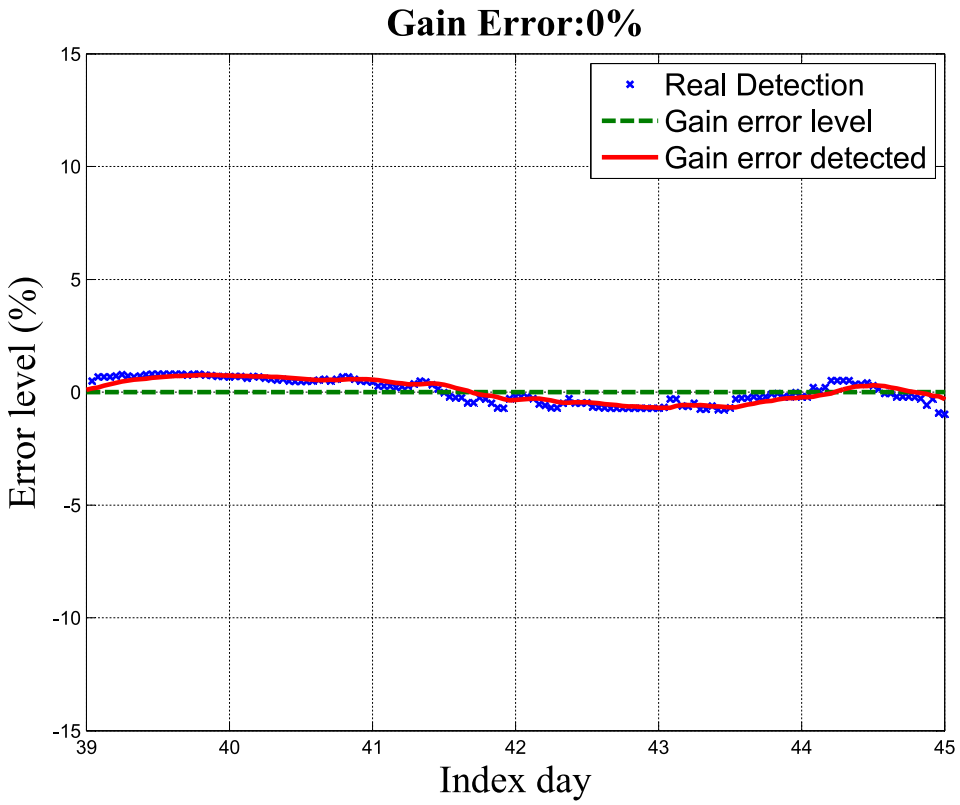
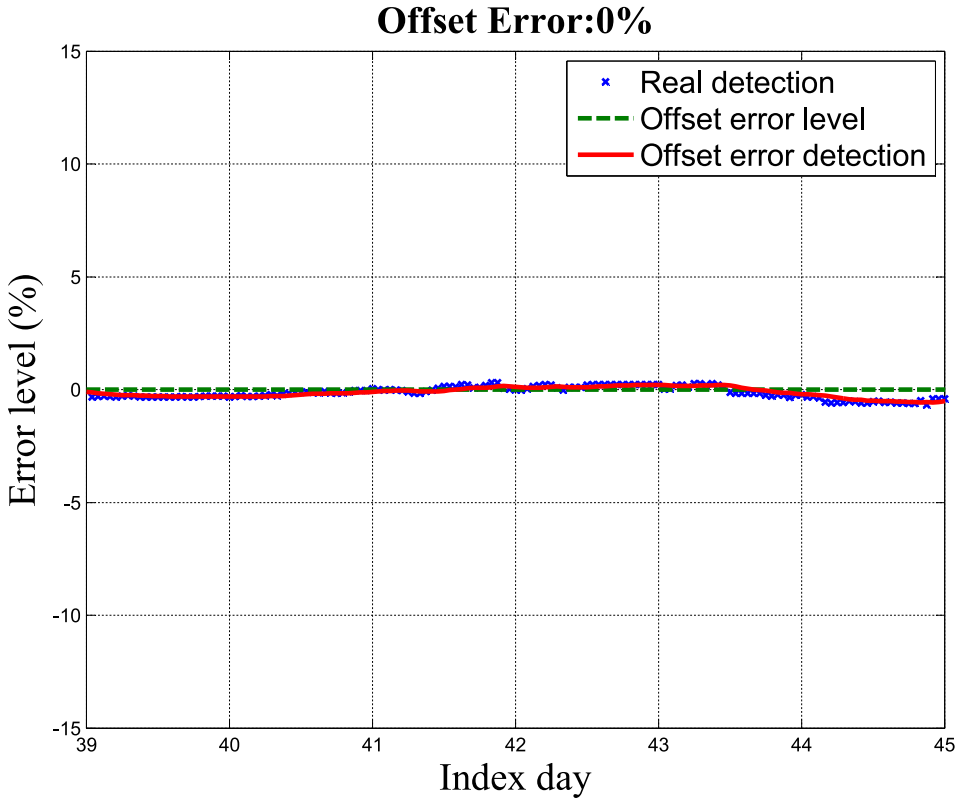


Figure 4.5: CT 1779 - Day Forecasting: Simulated load for the day after the training and forecast for that period are depicted.

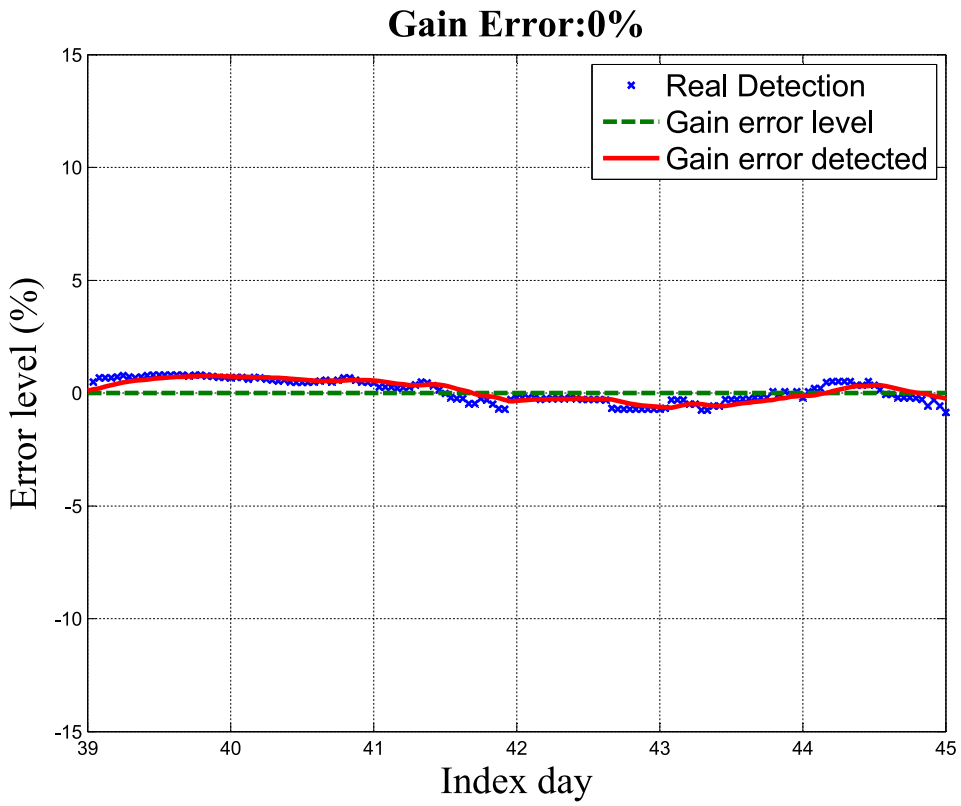


(a) Gain error detection

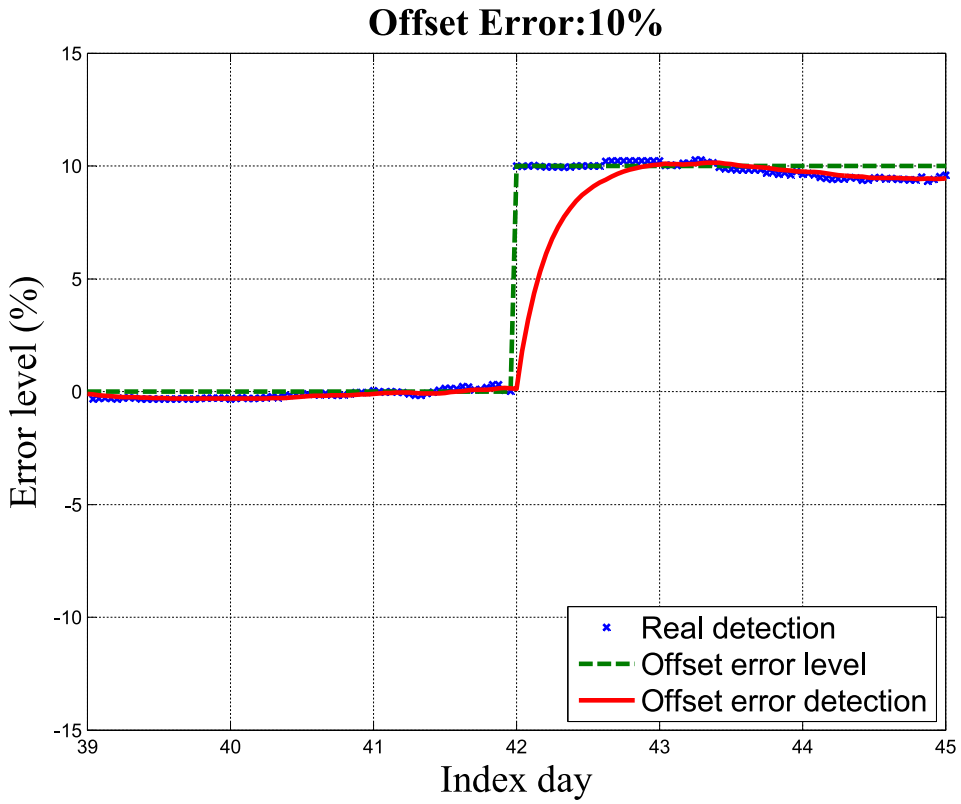


(b) Offset error detection

Figure 4.6: CT 1779 - Gain 0% & Offset 0%: Detecting 0% gain and 0% offset errors during the days after training.

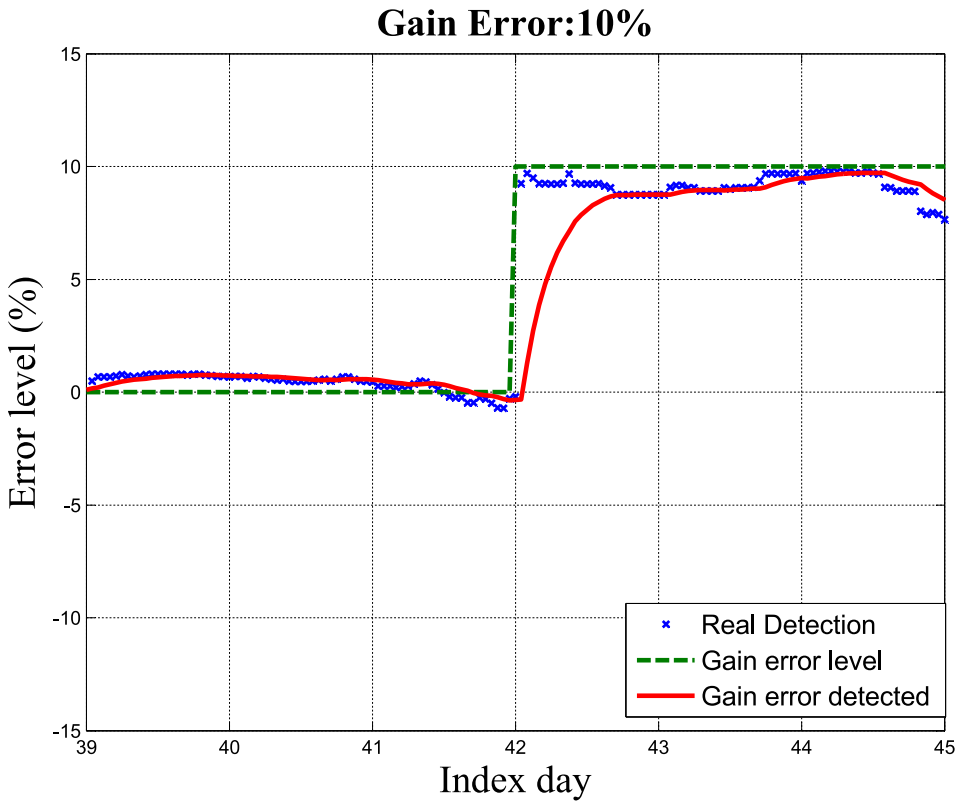


(a) Gain error detection

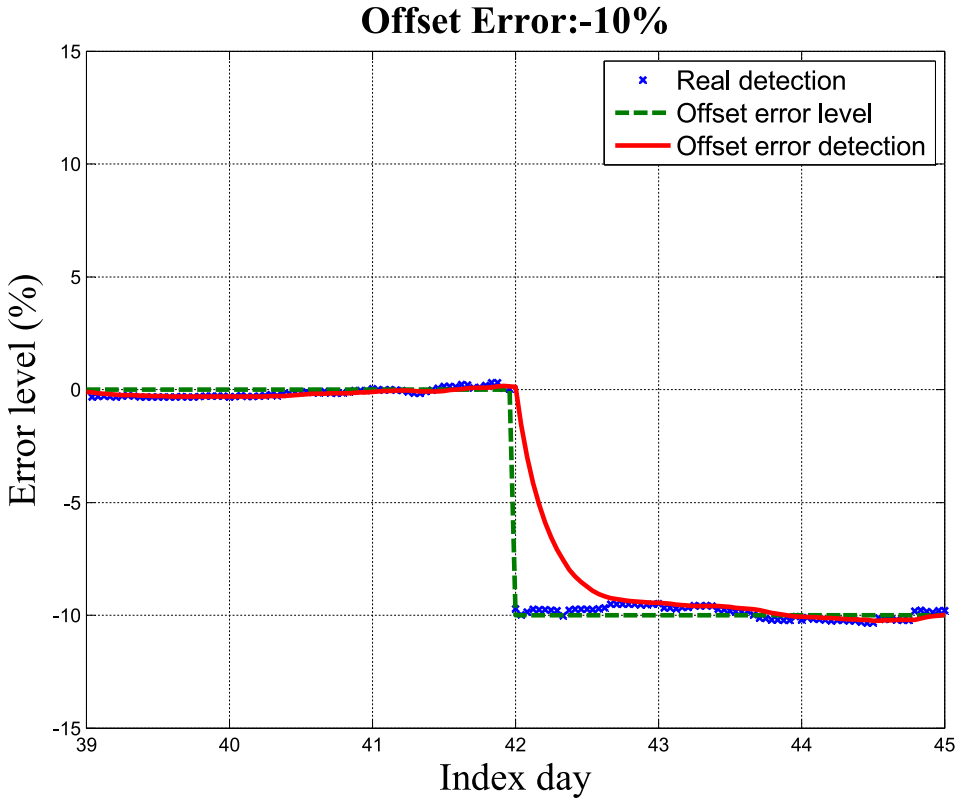


(b) Offset error detection

Figure 4.7: CT 1779 - Gain 0% & Offset 10%: Detecting 0% gain and 10% offset errors during the days after training.

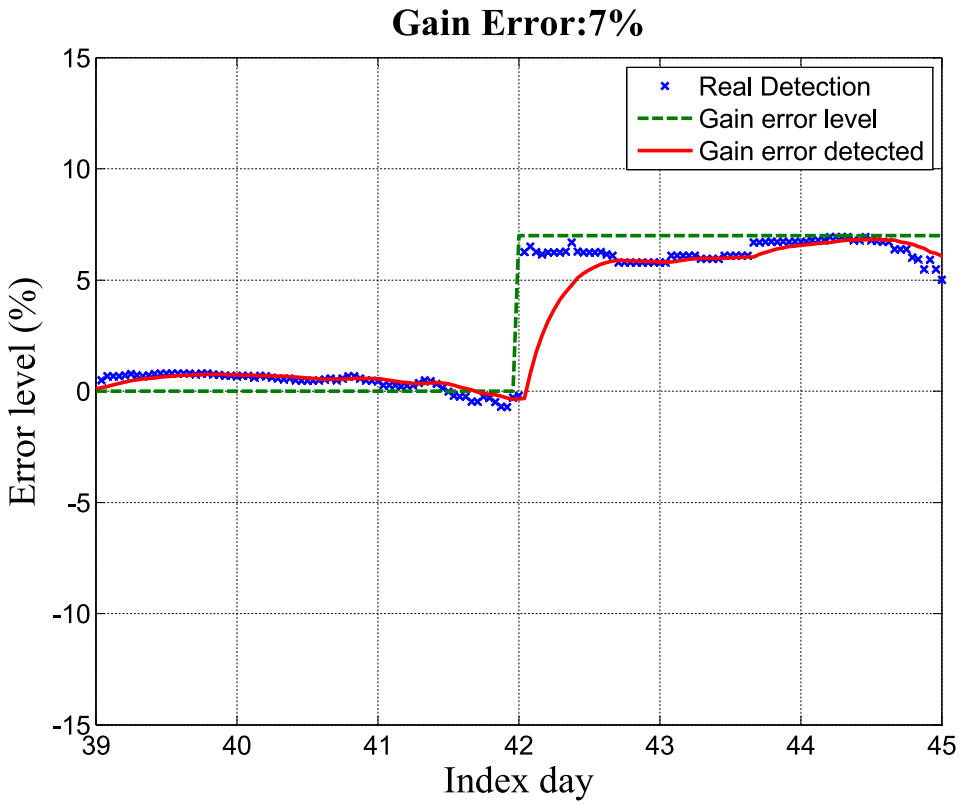


(a) Gain error detection

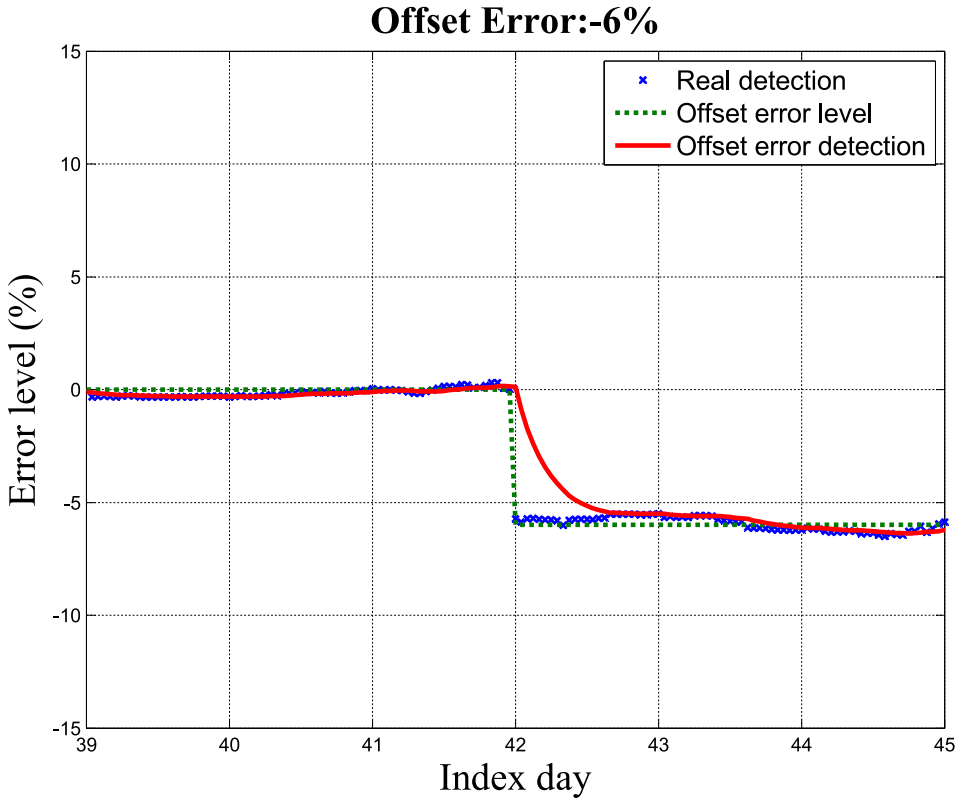


(b) Offset error detection

Figure 4.8: CT 1779 - Gain 10% & Offset -10%: Detecting 10% gain and -10% offset errors during the days after training.



(a) Gain error detection



(b) Offset error detection

Figure 4.9: CT 1779 - Gain 7% & Offset -6%: Detecting 7% gain and -6% offset errors during the days after training.

4.4.2. Real data

In the following figures, the results obtained from testing the described system against real data are depicted. Firstly, in figure 4.10, all the measurements taken from exactly 2 years (730 days) are shown. In this figure, it can be seen how the real data is not as good as the simulated one, some days do not have measurement for different kind of issues. In the cases where it was possible, the missing measurements were substituted by the measurements that were taken one year before.

Then, in figures 4.11, 4.12 and 4.13 it is shown a comparison between the forecast and the real measurements in different ways. Figures 4.11, 4.12 compare forecast and real measurements at the same hour during all available days. Finally, figure 4.13 depicts the 24 hourly loads forecast for the day d and the available measurements for that day. In these figures it can be seen a higher difference between the forecast and the real data, which gives a bigger forecast error and will increase the difficulty for detecting the measurement error.

Finally, there are several figures (4.14, 4.15, 4.16, 4.17) where the results with different error levels (offset and gain errors applied simultaneously in some cases) are shown. As there is a considerable forecast error in real data, the error detection will not be so accurate as it was in the simulated case.

As the prediction is worse in the real case, the error detection will give a worse result. In both cases, the output that shows the level of error has variation between the days although the included error is the same. That is the reason why the output of error level for all the different days will be approximated with the line that fits the best. This line will determine the error level that the EIT is experiencing.

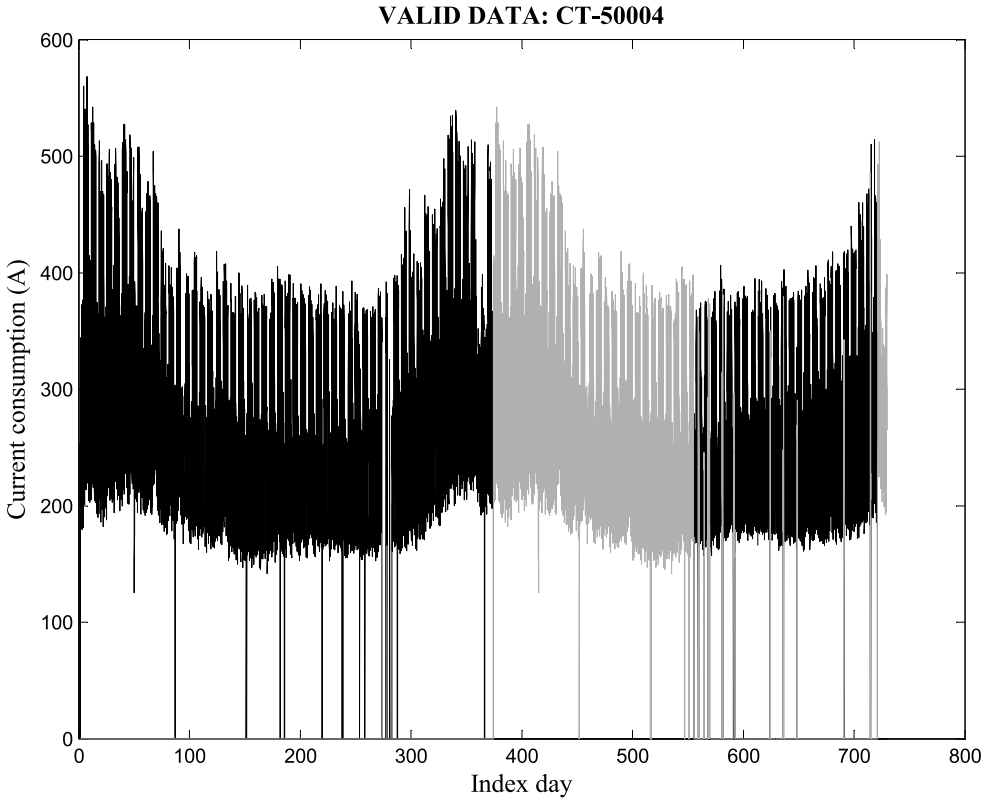


Figure 4.10: CT 50004 - two years measurements: Measurements taken during 715 days are depicted in this figure. In black real measured data and in gray the substituted data where there was missed data.

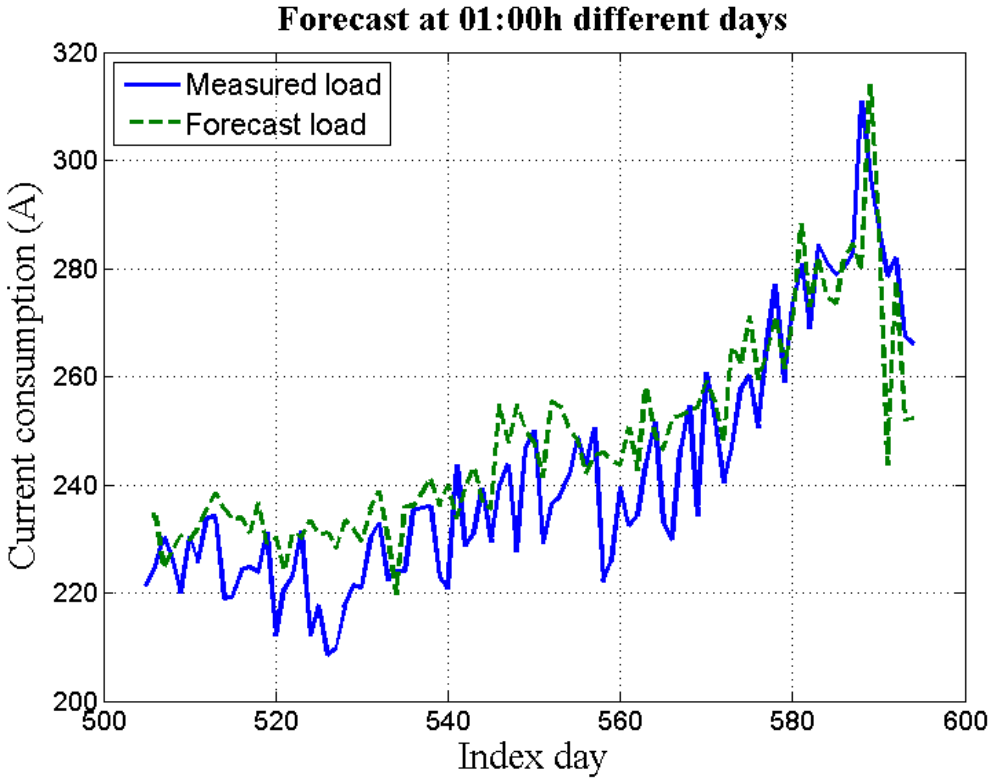


Figure 4.11: CT 50001 - Forecasting at 01.00h: During available days the measurement at 01:00h and the forecast for that hour are depicted.

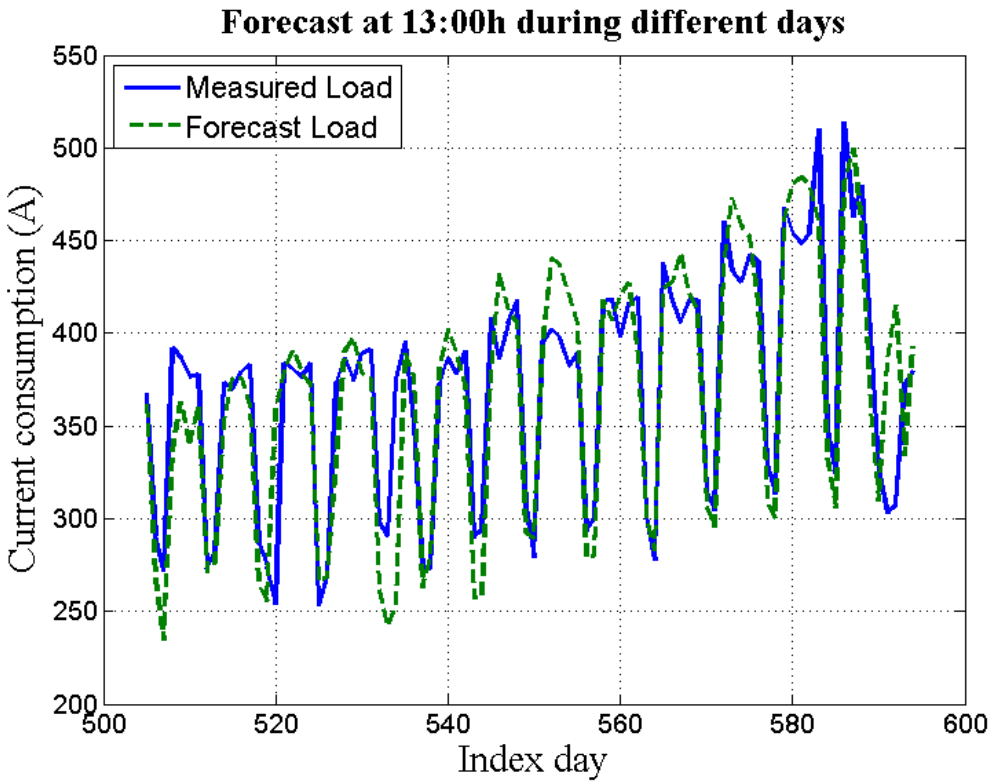


Figure 4.12: CT 50001 - Forecasting at 13.00h: During available days the measurement at 13:00h and the forecast for that hour are depicted.

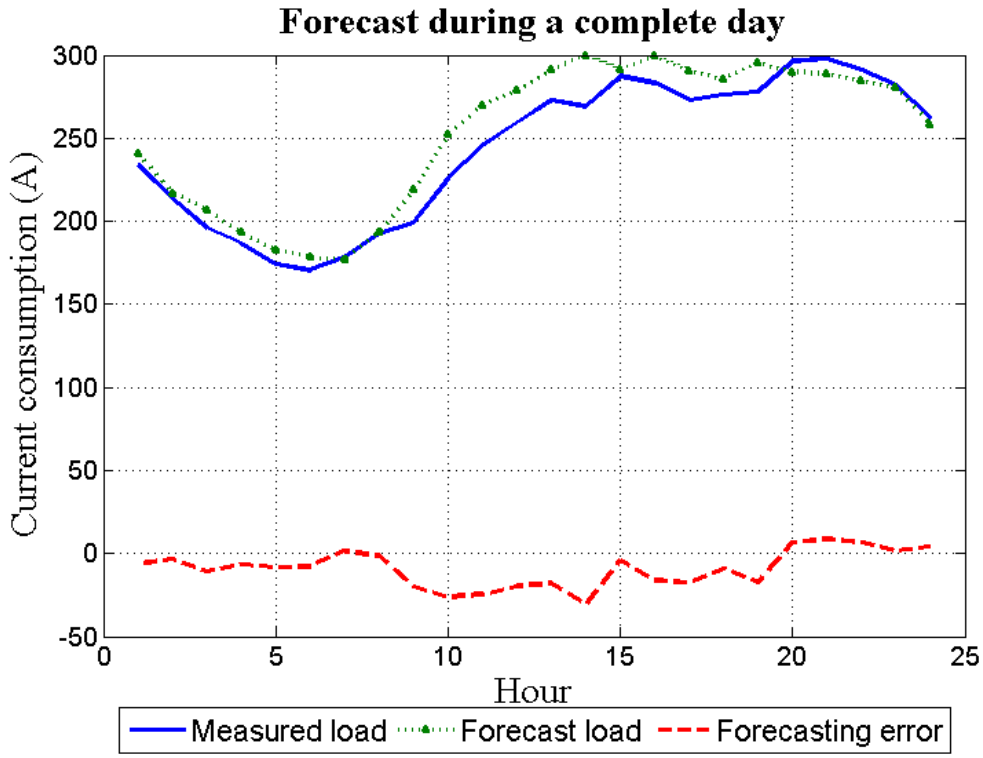
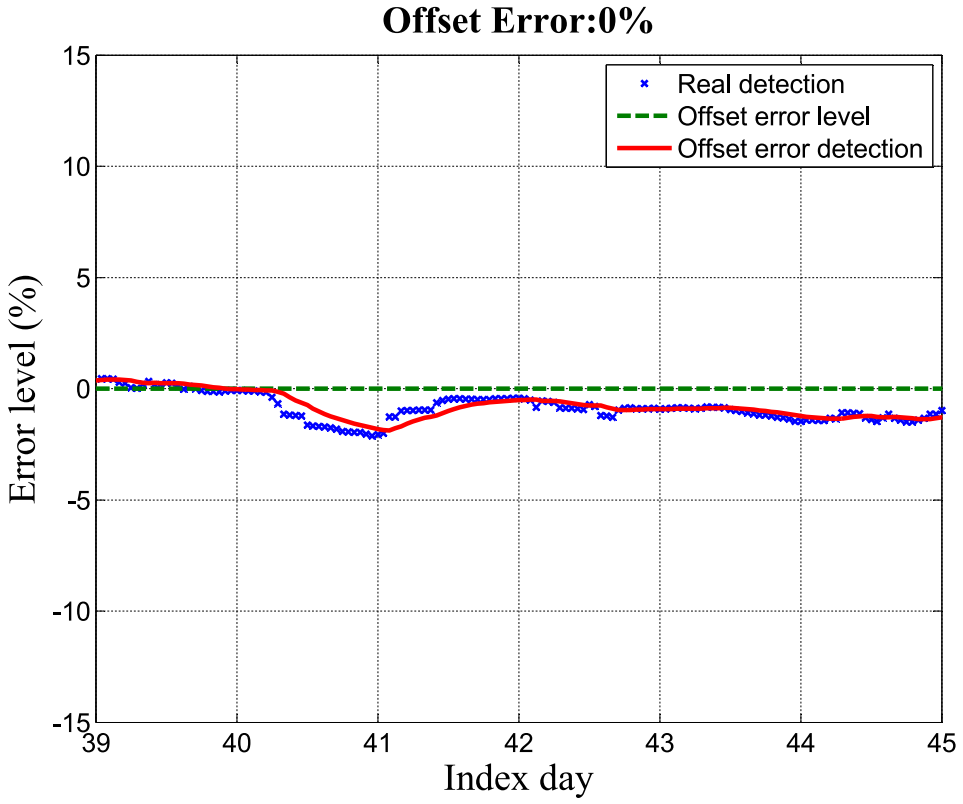


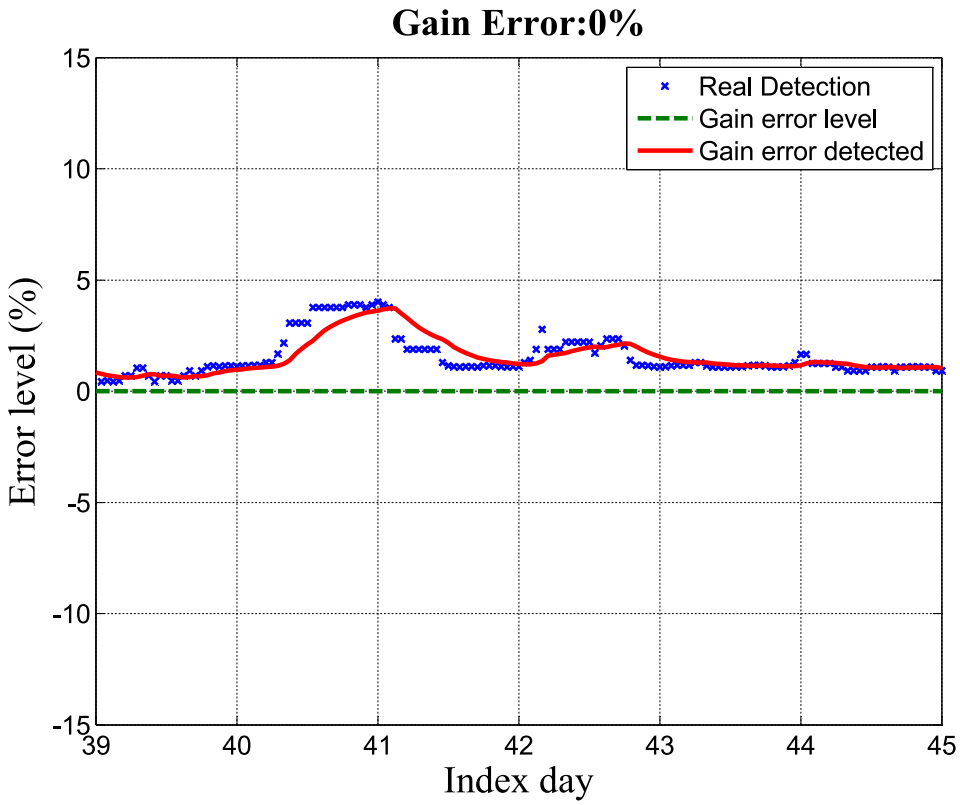
Figure 4.13: CT 1779 - Day Forecasting: Simulated load for the day after the training and forecast for that period are depicted.

(a) Gain error detection



(b) Offset error detection

Figure 4.14: CT 50001 - Gain 0% & Offset 0%: Detecting 0% gain and 0% offset errors during the days after training.

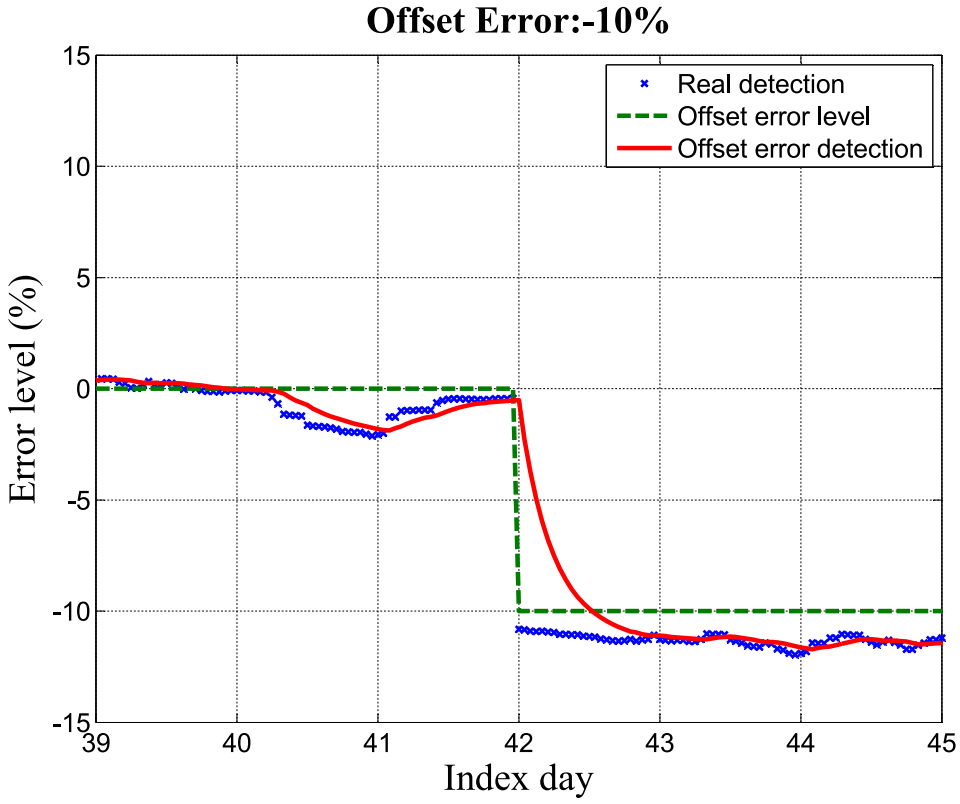


(a) Gain error detection

(b) Offset error detection

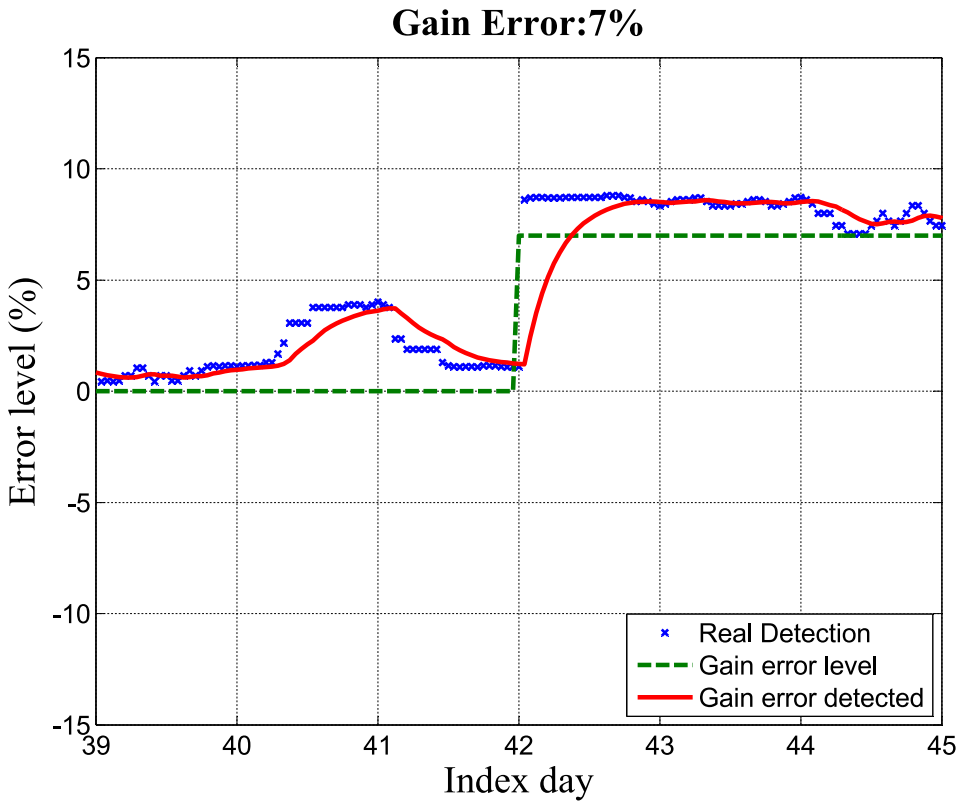
Figure 4.15: CT 50001 - Gain 0% & Offset 10%: Detecting 0% gain and 10% offset errors during the days after training.

(a) Gain error detection

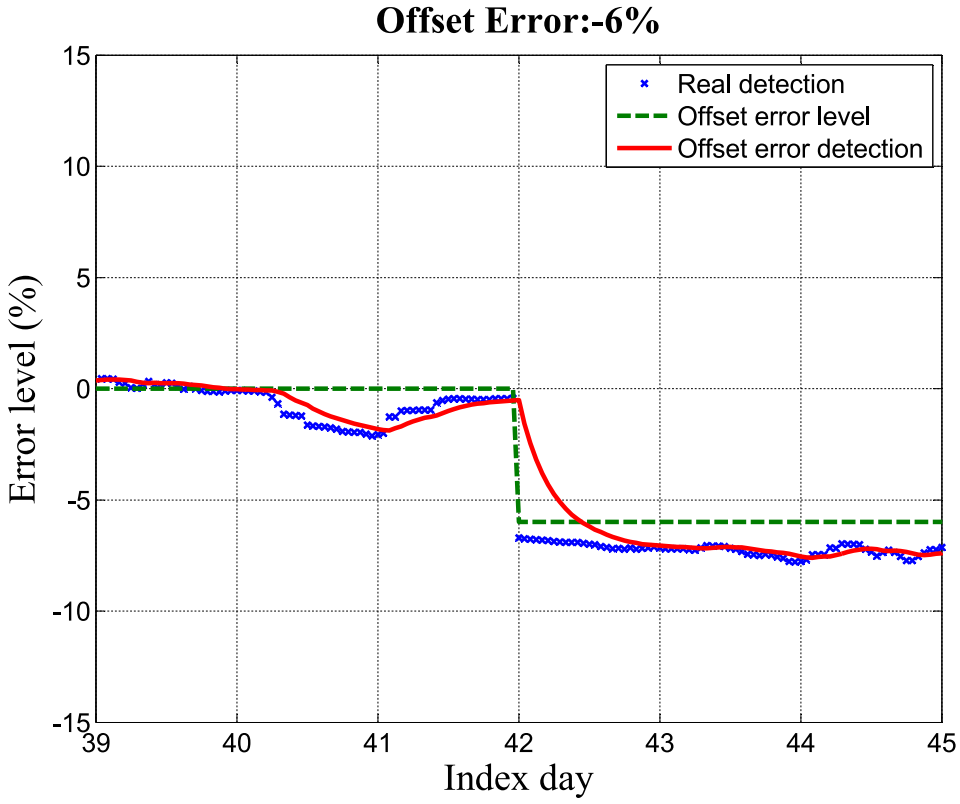


(b) Offset error detection

Figure 4.16: CT 50001 - Gain 10% & Offset -10%: Detecting 10% gain and -10% offset errors during the days after training.



(a) Gain error detection



(b) Offset error detection

Figure 4.17: CT 50001 - Gain 7% & Offset -6%: Detecting 7% gain and -6% offset errors during the days after training.

4.5. Analysis of the Results

For comparing the results that were obtained, an error analysis strategy has been implemented [36]. The MAPE, the NMBE and the NRMSE indicators are used to compare the results. These indicators show accuracy in terms of percentage and can be expressed as:

$$MAPE = \frac{1}{n} \cdot \sum_{i=1}^{n} \left| \frac{E_i - A_i}{E_i} \right| \cdot 100 \quad (\%) \tag{4.1}$$

$$NMBE = \frac{1}{n} \cdot \sum_{i=1}^{n} \left(\frac{E_i - A_i}{E_i} \right) \cdot 100 \quad (\%) \tag{4.2}$$

$$NRMSE = \sqrt{\frac{1}{n} \cdot \sum_{i=1}^{n} \left(\frac{E_i - A_i}{E_i} \right)^2} \cdot 100 \quad (\%) \tag{4.3}$$

where n denotes the number of samples, E stands for the error level injected and A represents the hourly detected error level.

Under normal operating conditions, the current measurement accuracy including all the equipment involved in the measuring process (mainly current transformer and measurement equipment) is below $\pm 2\%$. The contribution to the error of the transformer is not higher than $\pm 1\%$ [33] [34] and the measurement equipment [35] can introduce between $\pm 0.5\%$ and $\pm 1\%$ of error depending on the manufacturer.

Within this context, for ensuring that a systematic measurement offset or gain error is occurring without generating false alarms, a minimum threshold of $\pm 5\%$ is considered. Nonetheless, the results have been analyzed until a maximum measurement error of $\pm 10\%$.

MAPE and NRMSE indicators show that there exists considerable variability in the result. However, the NMBE indicator is smaller, showing that the estimated error level oscillates around the right value. As a consequence, by using a recursive least squares algorithm with forgetting factor, the measured error estimation is filtered. Likewise, a comparison between the current and the filtered estimated errors in absolute value is drawn:

$$abs = \left| E - \overline{F} \right| \tag{4.4}$$

where abs represents the absolute estimation error, E stands for the injected error level and $\overline{F}$ is the mean value of the filtered estimated error level along the different tests. Table 4.1 and 4.2 summarize the MAPE, NMBE, NRMSE and absolute errors indicators for gain and offset error in simulated data, respectively. Table 4.3 and 4.4 show the same indicators for real data.

In order to carry out this test and in an attempt to enhance the versatility of this system, simulated and real data from five EITs located in different SSs is used. In this regard, 20 different tests are set for each EIT, which involves training a new ANN for each test.

Finally, with the aim of showing average results for each studied error under simulated and real scenarios, an averaging process over the tests and over the different EITs is performed.

Table 4.5 details the results obtained when both offset and gain errors occur simultaneously. In the author's view, these results represent an excellent initial step toward the implementation of the system in open substation nodes for protection and security purposes.

Table 4.1: Error indicators for different gain error levels - Simulated data

	Simulated Data - Gain			
Error Level	**MAPE (%)**	**NMBE (%)**	**NRMSE (%)**	**Abs (%)**
-10 %	20,915	17,021	25,857	1,705
-9 %	21,629	16,357	27,001	1,472
-8 %	22,871	15,718	28,786	1,264
-7 %	24,970	15,353	31,574	1,076
-6 %	27,958	14,930	35,393	0,895
-5 %	32,548	14,694	41,157	0,741
+5 %	34,706	11,770	42,908	0,586
+6 %	29,723	11,694	36,719	0,697
+7 %	26,193	11,641	32,346	0,814
+8 %	23,615	11,558	29,150	0,920
+9 %	21,613	11,508	26,644	1,039
+10 %	20,068	11,513	24,702	1,152

Table 4.2: Error indicators for different offset error levels - Simulated data

	Simulated Data - Offset			
Error Level	**MAPE (%)**	**NMBE (%)**	**NRMSE (%)**	**Abs (%)**
-10 %	7,327	0,595	9,155	0,073
-9 %	8,146	0,903	10,170	0,085
-8 %	9,170	1,243	11,442	0,103
-7 %	10,486	1,662	13,078	0,122
-6 %	12,241	2,208	15,256	0,129
-5 %	14,664	2,839	18,269	0,140
+5 %	14,870	3,845	18,732	0,195
+6 %	12,487	3,354	15,742	0,202
+7 %	10,761	2,955	13,573	0,207
+8 %	9,455	2,633	11,931	0,211
+9 %	8,426	2,362	10,636	0,212
+10 %	7,597	2,132	9,592	0,213

Table 4.3: Error indicators for different gain error levels - Real data

	Real Data - Gain			
Error Level	**MAPE (%)**	**NMBE (%)**	**NRMSE (%)**	**Abs (%)**
-10 %	29,476	11,584	35,924	1,156
-9 %	32,658	11,662	39,666	1,047
-8 %	36,603	11,560	44,344	0,925
-7 %	42,024	11,582	50,772	0,809
-6 %	48,871	11,561	58,913	0,690
-5 %	58,561	11,506	70,410	0,574
+5 %	62,436	12,890	74,999	0,641
+6 %	52,737	13,088	63,483	0,784
+7 %	45,686	13,172	55,110	0,924
+8 %	40,366	13,091	48,779	1,045
+9 %	36,504	13,218	44,210	1,188
+10 %	33,438	13,407	40,601	1,344

4.6. Conclusions

Regarding to the obtained results and their analysis, it is possible to obtain some conclusions that have been previously described and now they will be highlighted.

Firstly, it should be noted that there is a big difference between the simulated and the real data

Table 4.4: Error indicators for different offset error levels - Real data

	Real Data - Offset			
Error Level	**MAPE (%)**	**NMBE (%)**	**NRMSE (%)**	**Abs (%)**
-10 %	12,932	0,568	16,230	0,072
-9 %	14,365	0,621	18,026	0,072
-8 %	16,153	0,681	20,268	0,071
-7 %	18,448	0,750	23,143	0,069
-6 %	21,503	0,829	26,968	0,066
-5 %	25,773	0,919	32,313	0,064
+5 %	25,576	3,271	31,930	0,146
+6 %	21,472	3,053	26,788	0,168
+7 %	18,517	2,822	23,090	0,183
+8 %	16,282	2,600	20,295	0,192
+9 %	14,527	2,394	18,103	0,198
+10 %	13,111	2,209	16,337	0,000

Table 4.5: Combined error detection in real data

		Offset Error Level													
		-10%		**-7.5%**		**-5%**		**0%**		**5%**		**7.5%**		**10%**	
Gain Error Level	**-10%**	-8.9	-10.3	-8.9	-7.8	-8.9	-5.3	-8.8	-0.4	-8.5	4.5	-8.5	7.0	-8.5	9.5
	-7.5%	-6.7	-10.2	-6.7	-7.7	-6.7	-5.2	-6.6	-0.3	-6.3	4.6	-6.3	7.1	-6.3	9.5
	-5%	-4.5	-10.1	-4.5	-7.6	-4.5	-5.1	-4.4	-0.2	-4.1	4.7	-4.1	7.1	-4.1	9.6
	0%	-0.1	-9.9	-0.1	-7.4	-0.1	-4.9	0.0	0.0	0.2	4.9	0.3	7.4	0.3	9.8
	5%	4.2	-9.7	4.2	-7.2	4.2	-4.7	4.4	0.2	4.6	5.1	4.6	7.5	4.7	10.0
	7.5%	6.4	-9.6	6.4	-7.1	6.4	-4.6	6.5	0.3	6.8	5.2	6.8	7.6	6.8	10.2
	10%	8.5	-9.5	8.5	-6.7	8.6	-4.5	8.7	0.4	8.9	5.3	9.0	7.8	9.0	10.3

Gain error level detected Offset error level detected

results. This difference is not only due to the fact that the real load data is less correlated between the days than the simulated one, it is also due to the fact that there are lost measurements for different issues and it is very harmful to the system developed in this project.

Secondly, the results show that there exists a big variation between the detected measurement errors and the actual error as it is shown with the Mean Absolute Percentage Error indicator. This variability is caused because the forecast error also has a high variability between the days and that is why there exists a bigger MAPE in the real data than in the simulated one. However, it was found that if a same level of error is maintained during several days, the error measurement detection tends to variate or oscillate around the right level of error.

Finally, as a solution for the issue exposed above, approximating the measurement error detection to the line that fit the best and making a comparison between the line and the real error, it was obtained acceptable results for both simulated and real data with less than 1 % of absolute error detection.

In this way, it can be concluded that the system developed in this work, presents results that are suitable with the targets that were proposed at the beginning of this work.

Chapter 5

Conclusions and future works

5.1. Conclusions

In this project, a new system for detecting measurement errors in secondary substations has been implemented. In this way, it is possible to detect an EIT misbehaviour while it is connected and operating.

With the historical measurements taken from an EIT and implementing a model of short term load forecasting the forecast based on an Artificial Neural Network, the forecast measurements for the day (d) are obtained. Comparing these ones with the actual measurements, the offset and gain level errors can be obtained independently.

In this way, important information is obtained, letting the operator to know which device could be given bad measurements, and the verification tests could be realised for the substitution. Therefore, the EIT replacement process is improved.

The described system developed here has also been published in Sensors - Open Access Journal which belongs to the first quartile in Instruments and Instrumentation area [37].

5.2. Future work

This work could be continued in the following several ways:

Firstly, the better forecasting model is used, the less forecast error will appear and a better measurement error detection will be obtained. In this way a comparison between different forecasting methods could be carried out.

Different errors could be try to be detected as well with the comparison between the forecast data and the actual ones, as an example, abnormal random error could be tried to be detected.

Finally, using different data, such as, temperature, wind speed, economic rates, etc. A better load forecast may be implemented and the error detection could be improved.

Chapter 6

Budget

This chapter will describe the theoretical cost of the whole project. It will include the material cost and the professional fees. Finally, the taxes will be added for getting the total cost of the project.

6.1. Material cost

In this section, the cost of the different materials (hardware and software) are detailed and the VAT (21%) is included.

Table 6.1: Material Costs (AVT included)

Item		**Unit Price (euro)**	**Units**	**Total Cost(euro)**
Hardware	Windows PC i7 2.93 GHz	600	1	600
Hardware total cost				600
Software	MATLAB	120	1	120
	ANN Toolbox (included in MATLAB)	0	1	0
	PCA Toolbox (included in MATLAB)	0	1	0
	LaTeX Editor	0	1	0
Software total cost				120
Material total cost				**720**

6.2. Professional fees

In this section the different Professional fees are calculated. This fees are calculated as gross incomes. It includes all the professional activities related with the project.

Table 6.2: Professional fees (gross salary)

Activity	Price (euro)	Time (months)	Total Cost (euro)
Engineering	1500	4	6000
Typing	1000	1	1000
Fees Total Cost			**7000**

6.3. Total cost

Table 6.3: Additional costs and total

Material Cost	720
Professional fees	7000
Printing	90
Transport	250
Total	**8060**

Bibliography

[1] D. Kriesel, *A Brief Introduction to Neural Networks*, 2007. [Online]. Available: http://www.dkriesel.com

[2] "Cs231n convolutional neural networks for visual recognition," http://cs231n.github.io/neural-networks-2/.

[3] J. a. Jiménez, M. Mazo, J. Ureña, A. Hernández, F. Álvarez, J. J. García, and E. Santiso, "Using PCA in time-of-flight vectors for reflector recognition and 3-D localization," *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 909–924, 2005.

[4] *Understanding data converters*. Texas instruments.

[5] "Adjusting and calibrating out offset and gain error in a precision dac," http://www.maximintegrated.com/en/app-notes/index.mvp/id/4602.

[6] M. Schumacher, C. Hoga, and J. Schmid, "Get on the digital bus to substation automation," *Power and Energy Magazine, IEEE*, vol. 5, no. 3, pp. 51–56, May 2007.

[7] S. Sheng, D. Xianzhong, W. Chan, and L. Zhihuan, "Erroneous measurement detection in substation automation system using ols based rbf neural network," *International Journal of Electrical Power & Energy Systems*, vol. 31, no. 7, pp. 351–355, 2009.

[8] K.-L. Ho, Y.-Y. Hsu, and C.-C. Yang, "Short term load forecasting using a multilayer neural network with an adaptive learning algorithm," *Power Systems, IEEE Transactions on*, vol. 7, no. 1, pp. 141–149, 1992.

[9] D. C. Park, M. El-Sharkawi, R. Marks, L. Atlas, M. Damborg *et al.*, "Electric load forecasting using an artificial neural network," *Power Systems, IEEE Transactions on*, vol. 6, no. 2, pp. 442–449, 1991.

[10] N. Amjady and F. Keynia, "A new neural network approach to short term load forecasting of electrical power systems," *Energies*, vol. 4, no. 3, pp. 488–503, 2011.

[11] S.-J. Huang and K.-R. Shih, "Short-term load forecasting via arma model identification including non-gaussian process considerations," *Power Systems, IEEE Transactions on*, vol. 18, no. 2, pp. 673–679, 2003.

[12] N. Amjady, "Short-term hourly load forecasting using time-series modeling with peak load estimation capability," *Power Systems, IEEE Transactions on*, vol. 16, no. 3, pp. 498–505, 2001.

[13] M. Gulin, M. Vasak, G. Banjac, and T. Tomisa, "Load forecast of a university building for application in microgrid power flow optimization," *ENERGYCON 2014 - IEEE International Energy Conference*, pp. 1223–1227, 2014.

[14] Z. Aung, M. Toukhy, J. R. Williams, A. Sanchez, and S. Herrero, "Towards Accurate Electricity Load Forecasting in Smart Grids," *The Fourth Int'l. Conf. on Advances in Databases, knowledge and data application*, no. c, pp. 51–57, 2012.

[15] L. Hernandez, C. Baladrón, J. M. Aguiar, B. Carro, A. J. Sanchez-Esguevillas, and J. Lloret, "Short-term load forecasting for microgrids based on artificial neural networks," *Energies*, vol. 6, no. 3, p. 1385, 2013. [Online]. Available: http://www.mdpi.com/1996-1073/6/3/1385

[16] J. Sousa, L. Neves, and H. Jorge, "Assessing the relevance of load profiling information in electrical load forecasting based on neural network models," *International Journal of Electrical Power & Energy Systems*, vol. 40, no. 1, pp. 85 – 93, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S014206151200035X

[17] S. Naidu, E. Zafiriou, and T. J. McAvoy, "Use of neural networks for sensor failure detection in a control system," *Control Systems Magazine, IEEE*, vol. 10, no. 3, pp. 49–55, April 1990.

[18] Q. Pengpeng, L. Jinguo, Z. Wei, and L. Yangmin, "Sensor fault diagnosis method based on fractal dimension," in *Control and Decision Conference (CCDC), 2013 25th Chinese*, May 2013, pp. 2955–2960.

[19] M. R. Napolitano, C. Neppach, V. Casdorph, S. Naylor, M. Innocenti, and G. Silvestri, "Neural-network-based scheme for sensor failure detection, identification, and accommodation," *Journal of Guidance, Control, and Dynamics*, vol. 18, no. 6, pp. 1280–1286, 1995.

[20] S. Sheng, D. Xianzhong, W. Chan, and L. Zhihuan, "Erroneous measurement detection in substation automation system using ols based rbf neural network," *International Journal of Electrical Power & Energy Systems*, vol. 31, no. 7-8, pp. 351 – 355, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0142061509000416

[21] S. Roweis, "Levenberg-marquardt optimization."

[22] S. K. Chidrawar, S. Bhaskarwar, and B. M. Patre, "Implementation of neural network for generalized predictive control: A comparison between a newton Raphson and Levenberg Marquardt implementation," *2009 WRI World Congress on Computer Science and Information Engineering, CSIE 2009*, vol. 1, pp. 669–673, 2009.

[23] T. Jia-Li, L. Yi-Jun, and W. Fang-Sheng, "Levenberg-marquardt neural network for gear fault diagnosis," in *Networking and Digital Society (ICNDS), 2010 2nd International Conference on*, vol. 1. IEEE, 2010, pp. 134–137.

[24] I. Jolliffe, *Principal component analysis*. Wiley Online Library.

[25] L. I. Smith, "A tutorial on principal components analysis," Cornell University, USA, Tech. Rep., February 26 2002. [Online]. Available: http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf

[26] D. L. Swets, "Using discriminant eigenfeatures for image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 831–836, 1996.

[27] Y. Mami and D. Charlet, "Speaker identification by anchor models with pca/lda post-processing," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, vol. 1. IEEE, 2003, pp. I–180.

[28] A. L. Rukhin, "Analysis of time series structure ssa and related techniques," *Technometrics*, vol. 44, no. 3, pp. 290–290, 2002.

[29] H. Hassani, "A brief introduction to singular spectrum analysis," 2010.

[30] D. Hofmann, "Common sources of errors in measurement systems." [Online]. Available: http://eu.wiley.com/legacy/wileychi/hbmsd/pdfs/mm154.pdf

[31] S. Salcedo-Sanz, A. M. Pérez-Bellido, E. G. Ortiz-García, A. Portilla-Figueras, L. Prieto, and F. Correoso, "Accurate short-term wind speed prediction by exploiting diversity in input data using banks of artificial neural networks," *Neurocomputing*, vol. 72, no. 4-6, pp. 1336–1341, 2009.

[32] W.-c. Yu and N.-y. Shih, "Bi-Loop Recursive Least Squares Algorithm," vol. 13, no. 8, pp. 505–508, 2006.

[33] T. Yamada, S. Kon, N. Hashimoto, T. Yamaguchi, K. Yazawa, R. Kondo, and K. Kurosawa, "ECT evaluation by an error measurement system according to IEC 60044-8 and 61850-9-2," *IEEE Transactions on Power Delivery*, vol. 27, no. 3, pp. 1377–1384, 2012.

[34] W. Li, S. Zhang, W. Ma, and M. Ao, "Application limitation of electronic current and Voltage Transformers in digital substations," *PEAM 2011 - Proceedings: 2011 IEEE Power Engineering and Automation Conference*, vol. 2, pp. 496–499, 2011.

[35] ZIV grid automation, "7IRD-E Distribution Protection & Control Terminal," http://www.gridautomation.ziv.es/doc-downloads/literature/manuals/B7IR1107Ev00.pdf.

[36] M. G. De Giorgi, P. M. Congedo, M. Malvoni, and D. Laforgia, "Error analysis of hybrid photovoltaic power forecasting models: A case study of mediterranean climate," *Energy Conversion and Management*, vol. 100, pp. 117–130, 2015.

[37] J. Moriano, F. J. Rodríguez, P. Martín, J. A. Jiménez, and B. Vuksanovic, "A new approach to detection of systematic errors in secondary substation monitoring equipment based on short term load forecasting," *Sensors*, vol. 16, no. 1, p. 85, 2016.

Appendix A

User guide

A.1. Introduction

A graphical user interface was developed for easing the operator handling. The GUI was developed employing the MATLAB GUI Layout Editor and it is compatible with both simulated and real data. This interface is divided in five parts that are depicted in figure A.2. Every part is now explained:

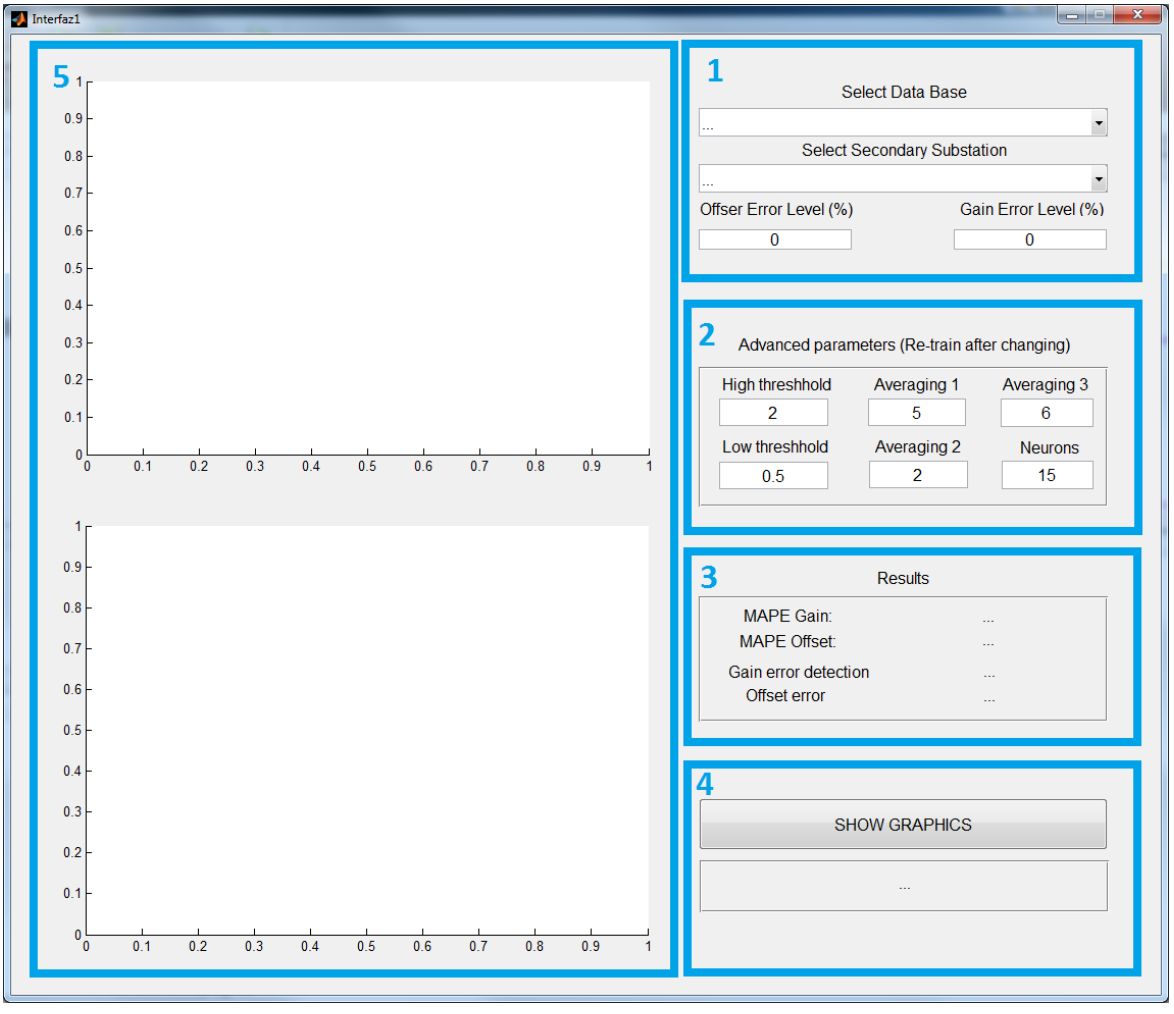


Figure A.1: Interface parts depiction

1. In this part, the data base (Real or simulated) is selected, depending on the chosen data base, different secondary substations will appear in the second pop up menu for being picked. At this point, the system is trained and once that the train is finished different offset and gain error levels can be introduced.

2. Here, different advanced parameters can be modified. As the modification affects the results of the training, after modifying any parameter it is needed to retrain the network loading new data. Every parameter will be explained in more detail in section A.2.

3. In this panel the results will be shown and the value of the performance will be shown. This result compares the actual error value with the estimated error value by different processes that are explained in section A.2.

4. A button is available for showing the last created graphic whenever it is wanted to be seen and a dialogue box where the program will return information of how the process is going.

5. Finally, when it is required, the graphics will be shown in the two axis available on the left part. The top axis contains depicts the gain error and the bottom one shows the offset error.

A.2. Manual

For running the interface, select "Interfaz1" as current folder in MATLAB and write "Interfaz1" in the command window. The GUI will be as depicted in A.2.

Then it is needed to select one data base. Two data base are available:

- Simulator: This data base is composed by simulated data, and **three** different secondary substations will be available.
- Real data: This data base is composed by real data, and **nine** different secondary substations will be available for analysis.

One of the available secondary substations needs to be selected and internally the system will be trained. The data of the substation is shown and once that the process is finished, the dialogue box will shown the message "Done".

At this point, the graphics and the result for 0% gain error and 0% offset error can be depicted by clicking on the "Show graphics" button.

If we are not happy with the result, the different advanced parameters can be modified until getting an acceptable result. The different parameters needs to be explained:

- High threshold: It determines the maximum acceptable gain value for an hour expressed as parts per unit, in case that the obtained value is higher than this threshold it will not be included for the gain level detection.
- Low threshold: It determines the minimum acceptable gain value for an hour expressed as parts per unit, in case that the obtained value is lower than this threshold it will not be included for the gain level detection.
- Averaging 1: This parameter is used for select the number of hourly samples to average before the differentiation process.

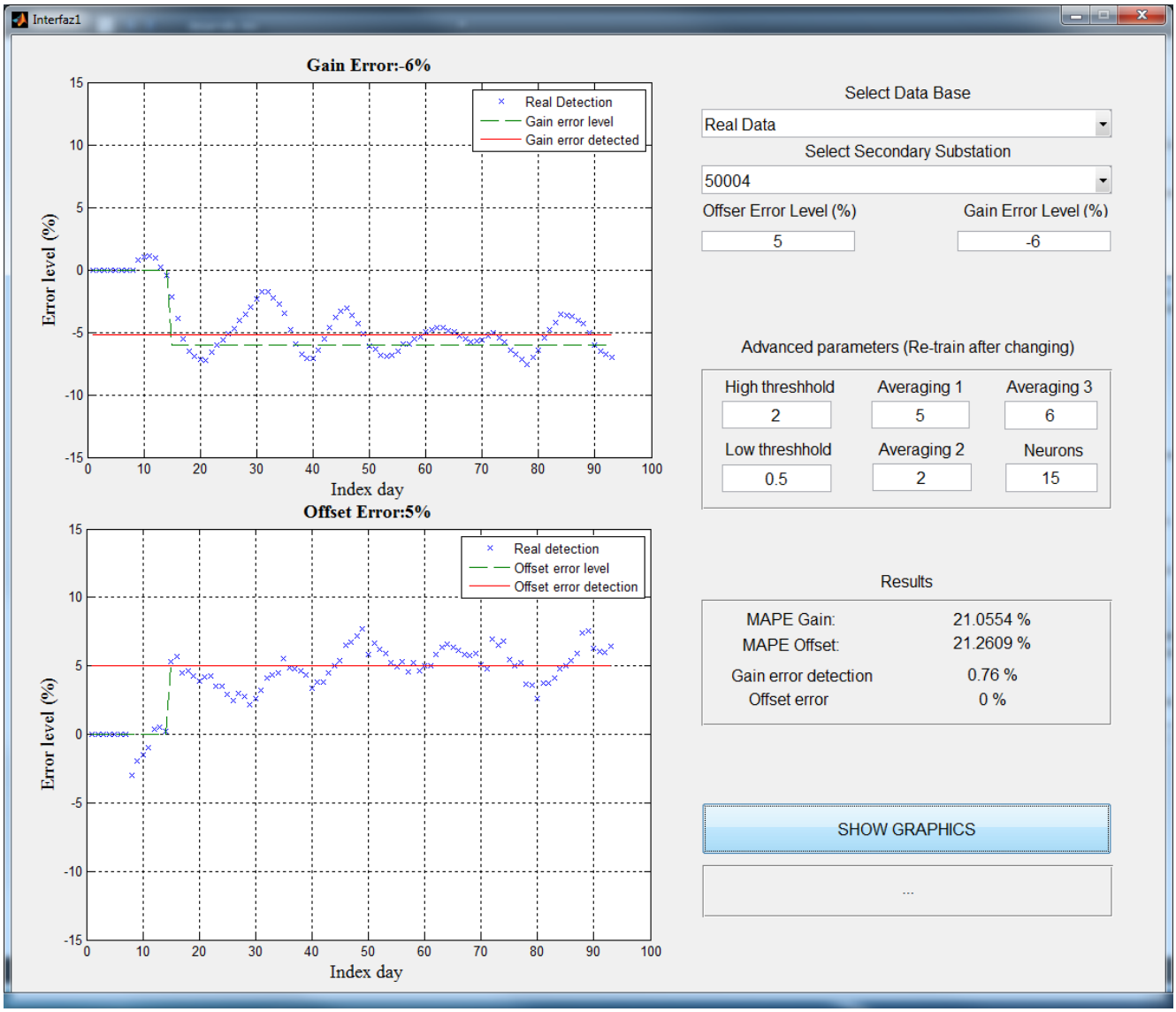


Figure A.2: Example of interface working

- Averaging 2: It indicates the number of averaging samples once that the differentiation has been carried out.
- Averaging 3: This parameter indicates the number of days that are averaged for determine the gain value.
- Neurons: This parameter allows the user to determine how many neurons will be in the hidden layer of the ANN. It is recommended to use a value between 10 and 20 neurons.

Appendix B

Specifications

For this project, the following specifications need to be satisfied:

- Computer with approximately 4 Gb of free space. The minimum requirements are not specified but for a proper work of the program it will be necessary a powerful computer. In this case a Windows PC i7 2.93 GHz was employed.
- The following software was needed for completing the project:
  1. MATLAB R2014a with the ANN and PCA toolboxes
  2. LaTeX editor

Universidad de Alcalá
Escuela Politécnica Superior

ESCUELA POLITECNICA
SUPERIOR

Universidad
de Alcalá