FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Load Forecasting on Special Days

Daniel Carlos Silva Costa e Sá

Mestrado Integrado em Engenharia Electrotécnica e de Computadores

Supervisor: Prof. Dr. Vladimiro Miranda

Second Supervisor: Dr. Jean Sumaili

Porto, July 2013

U.PORTO
FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

MIEEC - MESTRADO INTEGRADO EM ENGENHARIA ELETROTÉCNICA E DE COMPUTADORES | **2012/2013**

A Dissertação intitulada

"Load Forecasting on Special Days"

foi aprovada em provas realizadas em 16-07-2013

o júri

Peças Lopes

Presidente Professor Doutor João Abel Peças Lopes
Professor Catedrático do Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto

Professor Doutor José Boaventura Ribeiro da Cunha
Professor Associado c/ Agregação do Departamento de Engenharia Electrotécnica e de Computadores da Universidade de Tràs os Montes e Alto Douro

V. Miranda

Professor Doutor Vladimiro Henrique Barrosa Pinto de Miranda
Professor Catedrático do Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto

Doutor Jean Sumaili
Investigador do INESC - TEC

O autor declara que a presente dissertação (ou relatório de projeto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extratos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são corretamente citados.

Daniel Carlos Silva Costa e Sá
Autor - Daniel Carlos Silva Costa e Sá

Faculdade de Engenharia da Universidade do Porto

Resumo

Hoje em dia, com a desregulamentação do sistema de energia, a necessidade de maior eficiência e estabelecimento de novos padrões de preservação do meio ambiente foram introduzidas restrições mais duras sobre o planeamento, gestão e controlo do sistema de energia.

O sucesso comercial das empresas de energia depende da capacidade de apresentar propostas competitivas, assim sendo, alcançar melhorias na previsão de carga pode levar a um aumento substancial dos lucros comerciais. Deste modo, existem muitos métodos de previsão que têm sido publicados na literatura científica, cada um deles com especificações diferentes, dependendo dos seus objetivos.

Nesta dissertação será levado em conta a repercussão com dias especiais, como feriados. A média dos erros de previsão de carga para os feriados é muito mais elevada em comparação com os dias normais, devido ao facto de nestas situações não existir quantidade suficiente de informação histórica para representar as suas características.

Várias técnicas de previsão têm sido aplicada a este tipo de previsão, a maioria das abordagens baseiam-se em técnicas de redes neuronais. Muitos investigadores têm apresentado bons resultados e melhorias visíveis em novas metodologias em comparação com métodos tradicionais, mas nenhum deles tem sido capaz de resolver o problema da falta de informação histórica em dias especiais. Portanto, esta tese tem como objetivo principal a resolução deste problema.

Nesta dissertação será feito o estudo de uma nova abordagem técnica para este problema, com base em Redes Neuronais Autoassociativas / Autoencoders como um estimador de dados em falta, em que serão considerados os dias especiais como os dados em falta.

O algoritmo *Information Theoretical Learning Mean Shift* é utilizado para um processo denotado *truque de densificação*, ou seja, preencher com dados virtuais um conjunto escasso de dados relacionados com o consumo de energia diária em dias especiais. Isto permite um treino adequado das redes neuronais com os dados virtuais, reservando-se todos os dados reais (escassos) para fins de validação.

Este método foi aplicado num problema de previsão de demanda com dados reais de uma concessionária de distribuição no Brasil, onde a previsão para dias especiais foi difícil devido à falta de dados em registos históricos.

Palavras-chave: Mean shift, Information Theoretic Learning, Rede Neuronal Autoassociativa, Autoencoder, previsão de carga, dias especiais, feriado.

Abstract

Nowadays with the deregulation of the power system, requirement of higher efficiency and establishment of new standards on environmental preservation, were introduced harder constraints on the planning, management and control of the power system.

Commercial success of the energy companies depends on the ability to submit competitive bids, and improvements in forecasting the load can lead to substantial increases in trading profits.

Therefore, exist many forecasting methods that have been published in scientific literature, each of them with different specifications, depending on its objectives.

In this dissertation the repercussion of some special days will be taken in consideration, such as holidays. Average load forecasting errors on holidays is much higher than those for normal days because in this situation there is not enough historical information to represent their characteristics.

Several forecasting techniques have been applied to this kind of forecasting, the majority of the approaches are based on neural network techniques. Many researchers have presented good results and visible improvements on new methodologies compared with the traditional methods but none of them has been able to solve the problem of the lack of historic information on special days. Therefore, this thesis have as its main purpose the resolution of this problem.

In this dissertation the studying of a new technical approach to this problem will be made, based in Autoassociative Neural Network (AANN) / Autoencoder as a missing data estimator, in which the special days will be considered as the input missing data.

The Information Theoretical Learning Mean Shift algorithm is used to a process nominated *densification trick*, i.e., populate, with virtual data, a scarce set related to daily energy consumption in special days. This allows the proper training of neural networks with the virtual data, reserving all the scarce real data for validation purposes.

This method was applied in a demand forecasting problem with real data of a Brazilian distribution utility, where the prediction for special days was difficult to be achieved due to the lack of data in historical records.

Keywords: Mean shift, Information Theoretic Learning, Autoassociative Neural Networks, Autoencoder, load forecasting, special days, holiday.

Agradecimentos

Agradeço ao meu orientador Prof. Vladimiro Miranda por todas as ideias inspiradoras, orientação e confiança, todos os conselhos e sugestões. Foi um privilégio trabalhar com ele no INESC Porto.

Quero agradecer toda a disponibilidade e prontidão dos colaboradores INESC Porto, ao Dr. Jean Sumaili que para mim foi desde logo um orientador e à Joana Hora e Vera Palma pela sua ajuda na compreensão de algumas das ferramentas necessárias para o desenvolvimento da tese.

Aos meus colegas e amigos que me apoiaram ao longo da minha formação académica, em especial ao Rodrigo e ao Rui.

Os meus agradecimentos à minha família, especialmente aos meus pais, António e Rita e aos meus irmãos, João e Francisco, por todo o apoio, incentivo e confiança em mim depositados ao longo desta jornada.

Aos meus amigos escuteiros do agrupamento 94, que complementaram a minha educação e transmitiram muitos dos valores que levo para a vida.

Uma palavra amável e gentil é dirigida à Rita, minha namorada e melhor amiga, por todo o amor e carinho, juntos, somos "aprendizes de viajante".

Agradeço a Deus pela vida, pela saúde e pelas bênçãos recebidas.

Daniel Sá

“Study the past
if you would define the future”

Confucius

Contents

List of Figures

List of Tables

Abbreviations and Symbols

List of abbreviations:

AANN	Auto Associative Neural Network
ANN	Artificial Neural Network
ARIMA	Autoregressive Integrated Moving Average
BNN	Bayesian Neural Network
EA	Evolutionary Algorithm
EC	Evolutionary Computation
EP	Evolutionary Programming
EPSO	Evolutionary Particle Swarm Optimization
ES	Evolutionary Strategy
FEUP	*Faculdade de Engenharia da Universidade do Porto*
GA	Genetic Algorithms
GBMS	Gaussian Blurring Mean Shift
GDP	Gross Domestic Product
GMS	Gaussian Mean Shift
INESC	*Instituto de Engenharia de Sistemas e Computadores*
ITL	Information Theoretic Learning
ITLMS	Information Theoretic Learning Mean Shift
LMA	Levenberg-Marquardt Algorithm
LP	Linear Programming
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MCC	Maximum Correntropy Criterion
MEE	Minimum Entropy Error
MEEF	Minimum Error Entropy with Fiducial Points
MLP	Multi Layer Perceptron
MSE	Mean Square Error
NMAE	Normalised Mean Absolute Error
NMSE	Normalised Mean Square Error
NN	Neural Network
PCA	Principal Component Analysis
pdf	probability density function
PSO	Particle Swarm Optimization
purelin	linear transfer function

RWN	Recurrent Wavelet Network
SCADA	Supervisory Control and Data Acquisition
SSE	Sum Square Error
std	standard deviation
STLF	Short Term Load Forecasting
tansig	hyperbolic tangent sigmoid transfer function
trainbr	bayesian regulation backpropagation
trainlm	Levenberg-Marquardt backpropagation

List of symbols:

σ	Parzen Window Size
λ	Lagrange multiplier
$\varphi(\cdot)$	Activation Function
G	Gaussian kernel
H	Renyi's quadratic entropy
V	*information potencial*
D_{CS}	Cauchy-Schwartz Distance
min	minimum

Chapter 1

Introduction

This master thesis was developed in INESC Porto, integrated in the Master Degree in Electrical and Computer Engineering at the Faculty of Engineering of the University of Porto (FEUP).

A new concept of load forecasting on special days is presented in this work, using the Information Theoretical Learning Mean Shift (ITLMS) algorithm in a process of densification of the real data set, resulting in the criation of virtual data to train an Autoassociative Neural Network (AANN) / Autoencoder.

The main concern is to resolve the problem of not enough amount of historical information to represent special days, such as holidays. This approach is based on Autoencoders as a missing data estimator, in which will be considered the special days as the input missing data. In order to predict the holiday demand, it is used the Evolutionary Particle Swarm Optimization (EPSO) as an optimization algorithm.

1.1 Background and Context

Load forecasting became an essential instrument in power system planning, management and operation.

The reasons for its growing importance are related to the deregulation of the power system. The energy market, demands higher efficiency and establishment of new standards on environmental preservation. This introduced harder constraints on power system management and control. These changes require more sophisticate tools of planning and operation and, therefore, more accurate predictions of load is necessary [7].

Nowadays, basic operating functions such as unit commitment, economic dispatch, hydro-thermal coordination, transaction evaluation, fuel scheduling, unit maintenance, transaction evaluation and system security analysis can be performed efficiently with an accurate and robust forecast, improving the security of the power system and reducing the generation and operation costs [8].

Commercial success depends on the ability to submit competitive bids, and improvements in forecasting the load can lead to substantial increases in trading profits. The forecasting tools are interesting not only to power system operators, but also to load serving entities, merchant plants or generators and other market participants [9].

The quest for top-quality forecasting involves a broad variety of investigation fields, including several areas of engineering, economy, meteorology, and others. This explains the considerable forecasting methods that have been published in scientific literature.

The practical details of each particular load forecasting implementation differ from case to case, depending on the objectives. In this dissertation the repercussion with special days, such as holidays will be taken into account.

Average load forecasting errors for the holidays are much higher than those for normal days. In fact, their rate of occurrence may be considerably higher when dealing with real data. Besides, these kinds of events may change the general forecasting operations, channeling the performance to unacceptable levels [7, 10]. Special days also make the load forecast more difficult to treat because in these situations there is not enough amount of historical information to represents their characteristics.

Various forecasting techniques have been applied to this type of forecasting and the majority of the recently reported approaches are based on neural network techniques. Many researchers have presented good results. The attraction for these methods lies in the assumption that neural networks are able to learn properties of the load, which would otherwise require careful analysis to discover [11].

In short, the motivation in this dissertation thesis lies in fact that special days have been a recurring problem in load forecasting and this new approach based on Autoencoders and ITLMS algorithm may come to represent a successful tool to resolve the problem which none of forecasting technique has been able to resolve, the scarce existence of historical information on these day's type.

1.2 Objectives

Considering the promising results given in the recent paper [4], Sumaili, Miranda et al. have applied a densification trick using Information Theoretical Learning Mean Shift algorithm to allow demand forecasting in special days with scarce data. This work will be explained in greater detail in the chapter 2.

In this dissertation will be studied this tool for scarce data treatment.

- Will the ITLMS algorithm be able to identify distinct clusters in consumption data using a process of clustering associating the holidays in distinct days of the week?

- Will the ITLMS algorithm be usefull to allow virtual data collecting of each distinct specific clusters?

- Will be needed the criation of more or less groups of virtual data in order to perform a suitable neural network training set?
- Will this tool be able to resolve the problem of lack of historical data on special days?

Consulting the relevant literature have been observed great results for Autoencoders used as recognition machine, with this powerful tool can be estimated missing data in a database.

- If it is considered the special days as a missing data, might this tool adequately predict these kind of days?
- Might Autoencoder be an usefull tool in a load forecasting?
- Comparing the achieved results with Autoencoders and the results of the work [4], which of them is the best?

Therefore this dissertation looking for achieve these objectives set out.

Chapter 2

State of the Art

Several research centers and companies invested in research and development of methods / models of load forecasting, which led to a large number of forecasting systems, some of which are already under operation and marketing. The prediction systems are essentially characterized by the forecast horizon (minutes, hours, days), computational complexity and value of the forecast error.

This chapter does not aim to present a detailed study of all forecasting methodologies in the literature, but rather to show that there are numerous applications developed with the aim of making load forecasting on special days.

The main factors influencing the load will also be presented in more detail, how the classification of load forecasting is performed and the kind of division of days per year.

2.1 Factors which influence the load behaviour

An electric network is formed by the random uniting of different consumers. Changes in the consumption of different groups makes up the future load different from the previous circumstances. It is therefore crucial to study and understand the factors which influence the load in order to present methods to minimize the difference between the actual load and the forecasted load.

There are many factors which influence the load behaviour. These factors all differ in terms of time of onset, duration and effect on the electricity consumption. According to [12], these factors may be divided into two groups, namely special events and ever-present factors. Sports events and strikes are examples of special events, while weather and human behavioural patterns are examples of ever-present factors.

The most significant external factor that influences the load is probably the weather [11, 13, 14, 15]. The factors relating to the weather that are usually taken into account are temperature, rainfall, humidity, wind speed and cloud cover, as it is logical that all these factors have an effect on the use of electricity. Other factors, such as the psychological effect of hot, sunny weather, air conditioners and television audience behavior have also been suggested. Others permanent factors that are related to the load behaviour are the state of the economy, the level of factories production and the GDP [8, 11, 12, 15, 16, 17, 18, 19, 20, 21].

As an example of special events we can regard blackouts, large customer behavior (e.g. a large enterprise is shut down), natural disasters (e.g. floods and earthquakes), national sports events or strikes. Other special events such as holiday periods, the Easter Weekend and the Christmas period also occur regularly and have a significant effect on the load [8, 15, 19, 22, 23].

The onset and duration of these factors are usually known in advance, however the exact effect that they will have on the consumption is not.

There are methods based on the relationship between some variables and the load. Observing these variables in time and their relationship with the load allows the projection of the load in the future and the forecasting of its behavior.

In order to reduce the forecast error on anomalous situations, some authors develop forecast techniques specialized on each of the factors (e.g. holidays, festivals, rapid weather changes) [14, 18, 19, 24, 25, 26, 27, 28, 29].

This explains, why there are so many research and applications of methods / models to improve the load forecasting.

2.2 Load forecasting classification

The load forecasting can be classified according to the future horizon of time, this temporal scope may have various limits depending on the purpose [11, 12, 20, 21, 30, 31].

There are three types of load forecasting:

- Long-term load forecasting which consists of forecasting the load demand curve from 1 year and is useful to project demand years ahead and help in strategic development such as scheduling construction of new power generators as well as the determination of prices and regulatory policy.

- Mid-term load forecasting which consists of forecasting the load demand curve from 1 week to 1 year, are mainly used as reference for studies of contingency, maintenance scheduling, and for negotiations to purchase, sale and exchange of energy between the agents of the electric system.

- Short-term load forecasting which consists of forecasting the load demand curve from 1 hour, 24 hour to one week ahead, and is essential for tasks such as the scheduling of fuel purchases, inter company power transactions, security analysis and short term maintenance scheduling.

In this thesis the short-term load forecasting applied to special days will only be approached.

2.3 Types of days per year

Normally, all days are divided into two principal groups named normal days (weekdays) and special days.

Holidays are special days that have a high influence in the load demand curve. In the data set, it is observed that load demand is lower on holidays than on normal days. Moreover, the load demand curve is not only affected on holidays, but also on days located before and after holidays.

According to [15], there are two types of special days, fixed by weekday and fixed by date. A special day fixed by weekday occurs always at the same weekday but its date varies. Its location within a year may even vary within a month (e.g. Easter), depending on the year and the special day. Special days fixed by date fall always at the same time of year (e.g. Christmas). However, as the date is fixed the weekday varies, and for example it may occur during the weekend in one year and in the middle of the week in the next year.

There may be different special days in different countries, regions and cities (e.g. the city day or the independence day).

Different researchers give different configurations to the week days.

In [8], it describes the normal days like days in which events such as national and religious celebrations or national and religious mourning ceremonies don't occur. Normal days may include any day of the week (Saturday, Sunday, Monday, Tuesday, Wednesday, Thursday, and Friday).

Special days are classified in three types, official holidays, days before official holidays, and days after official holidays:

- Official holidays (except for the two week New Year vacation) are days in which religious and national celebrations and mourning ceremonies occur.
- On days before holidays, there is less social activity and in mid-day there is a decrease in the consumption of electricity. Therefore for a more accurate forecast the days before official holidays should be studied in a different category.
- The load pattern of normal days after holidays is different from other normal days of the week and shows a decrease in the amount of electricity consumed especially during the early working hours of the day. Regarding the fact that this difference is caused by the official holiday, they are considered special days and should be studied in a different category.

In other papers [11], the classifying of the special days is not examined separately and all days are divided by the guiding principle of three distinct classes:

- Mondays-Fridays;
- Saturdays;
- Sundays.

In this type of configuration the special days are normally classified like Sundays [32, 33] but other researchers classified the special days like Fridays [34] or Saturdays [7] depending on the country or region under analysis.

Analysing another example, [11] the Taiwan power system (of 1986) suggest that the days should to be divided in four categories:

- Sundays and holidays;
- Mondays and days after holidays;
- Saturdays;
- weekdays except holidays.

The classification of the days of the week influence the model of load forecasting, so it is important to take in consideration what is the best configuration to achieve the objectives of the forecast.

2.4 Load forecasting on special days

The improvement of the accuracy of load forecasting is critical for increasing the reliability and efficiency of the power systems. The load forecasting problem is a complex nonlinear problem linked with social considerations, economic factors, and weather variations. In particular, load forecasting for holidays is a challenging task once only a small number of recent historical data is available, compared with what is available for normal weekdays and weekends.

According to [7], the practical details of each particular load forecasting implementation differ from case to case, depending mainly on forecasting objectives, prediction scope, variables to be predicted, historical data available (quantity and quality), and rate and repercussion of anomalous events. Anomalous events may adulterate the general forecasting operations, leading the performance to unacceptable levels.

So far, many studies of the load forecasting have been made, many of them developed for load forecast on special days and acceptable results have been achieved.

The load curves of the same special days are dissimilar each year due to the system load growth/decline trend [10, 35]. If this yearly growth is ignored, the general shapes of same days become similar. Therefore, for many studies of load forecasting of one special day, only the data of that day in previous years or special days with the same behavior (e.g. special days that occur on Wednesdays) is used.

With that in mind, the load forecasting methods can be divided into two principal categories: statistical methods and computational intelligence techniques [29].

In literature, statistical methods such as auto-regression and time series have been used broadly for STLF. A lot of models using classical techniques were created during the last decades, such as Box-Jenkins models, ARIMA models, Kalman filtering models, and the spectral expansion techniques-based models. All of these techniques work well in normal conditions, but they lead to incorrect results when dealing with special days. Extreme complicated relationships that lead to immense mathematical operations for load forecasting are one of the most important defects of these techniques. Time-consuming for load forecasting, intrinsic inaccuracy and numerical instability are another of their deficiencies [34].

In recent years, use of intelligent techniques based on neural networks have increased noticeably for solving engineering problems. For example, Artificial Neural Network (ANN) and Fuzzy systems are two powerful tools that can be used approximately in every prediction and modeling problem. It has been shown that ANN are universal approximators with the capability of modeling every nonlinear system [34]. Considering this capability, some researchers have designed ANN-based short term load forecaster. Contemporary load forecasting techniques, such as Similar-day approach, Bayesian Neural Networks (BNN), Recurrent Wavelet Network (RWN), Neural Network-Fuzzy methods, have been developed recently and showing more acceptable results than traditional methods.

Then some load forecasting models applied to special days will be presented.

2.4.1 Similar-day approach

The load forecasting on special days can be made based on similarity behavior of a holiday with another day of the week. In most of countries, the services affected by the occurrence of the holiday are similar to that which occur with weekends. For this reason, the load curve of a holiday is approximately equal to the weekends near this day. In some regions, this similarity is greater with Saturday, in others with Sunday and in others with Fridays. Some studies use this similarity with the holiday and another day of the week to proceed with the prediction. Fidalgo and Lopes [7] describe a solution to forecasting holidays to the Portuguese electrical system based on artificial neural networks. The holiday is provided through a neural network trained exclusively to the last Saturday. As the forecasting system knows only the behavior of Saturdays, this model will forecast holidays as if it were any Saturday. Other similar work was described by Srinivasan, Chang and Liew [33] where predictions of holidays based on similarity with Sundays were made. The author uses neural nets together with fuzzy logic to forecast in the region of Singapore.

In the paper [34] Barzamini et al. proved that the load pattern for official holidays is completely different for the load pattern for normal working days of the week, but has much similarity with the Friday nearest to it. Therefore, in this paper in order to forecast the load on official holidays for Iran National Power System and its regions is used the Friday neural network sub-model first, and after the primary forecast is corrected by applying the rules of the fuzzy-expert system.

Others researchers also applied this method in their model of load forecasting [11, 32, 36, 37, 38, 39].

2.4.2 Artificial neural networks

Within the most acclaimed tools in the field of load forecasting, the Artificial Neural Networks (ANNs) have been supplanting long-established techniques in many applications.

In part, this success may be justified by ANN advantages like its adapting capacity and its tolerance to noisy data [7]. In fact, it is rare to find papers that report a poorer performance of ANN when compared to other methods.

In particular, load forecasting represents one of the most successful ANN applications in the power system domain [22, 23, 17, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53].

An example of this tool is refered in [54] where a multi-layer artificial neural network with an adaptive learning algorithm is used to forecast system hourly loads up to 168 hours for the Public Utilities Board (PUB) of Singapore. The ANN-based load models are trained using hourly historical load data and daily historical maximum/minimum temperature data supplied by the PUB and Meteorological Service Singapore respectively. The models are trained by day types to predict daily peak and valley loads. The hourly forecast loads are computed from the predicted peak and valley loads and average normalized loads for each day type. The average absolute error for a 24-hour ahead forecast using the actual load and temperature data is shown to be 2.32% for Mondays through Sundays and 5.98% for ten special day types in a year.

However, conventional artificial neural networks (ANN) based short-term load forecasting techniques have limitations in their use on special days. This is due to dissimilar load behaviors of holidays compared with those of ordinary weekdays during the year and to insufficiency of training patterns. In this way, some researchers suggest new techniques with integrated models like Bayesian Neural Networks, Recurrent Wavelet Network or Neural Network-Fuzzy methods.

2.4.3 Bayesian Neural Networks

In the paper [35] Mahdavi et al. proposed a new short-term load forecasting method for special days in irregular load conditions. The proposed method uses a Bayesian neural network (BNN) to forecast the hourly loads of special days. To do that, hybrid Monte Carlo method was used. This type of learning enables to work with simpler architecture than in other works.

In this paper, instead of using hybrid methods, only the Multi Layer Perceptron (MLP) network for STLF was used. This architecture is used for forecasting all special days. The key point here is to apply Bayesian learning to train the MLP. This type of learning leading to a simpler architecture gets a better result in comparison with that of the previous work. As pointed out in [14] the benefits of Bayesian approach include an indication of the degree of uncertainty in the predictions, automatic selection of an appropriate scale for network weights, and the avoidance of overfitting. BNN can better deal with a limited data set.

The training patterns were collected from the historical load data for the years of 1996–2002. The method was tested with the actual load data of special days for the years of 2003–2004. The test results showed very accurate forecasting with the average percentage relative error of 1.93%.

Other researchers like [36, 55, 56, 57, 58] also use the Bayesian approach in its works.

2.4.4 Recurrent Wavelet Network

In the paper [59] Baniamerian et al. presented a dynamic model for short-term special days load forecasting which uses a Recurrent Wavelet Network (RWN).

The wavelet networks have been developed as a complex-neuron alternative for universal approximation. As an alternative, by increasing the complexity of network architecture (i.e. recurrent neural network) and further intricate problems can be tackled.

Based on the complexity of load series for special days and lack of data, it is reasonable to use recurrent wavelet network (RWN) [60]. This method has been rarely used for time series prediction [61], because of its initialization problem. Initialization method in RWN severely affects training process the same as wavelet network. Thus, a new initialization method is suggested, based on Orthogonal Least Square (OLS) technique. Moreover, this RWN, by back-propagation-based training, has been used for short-term load forecasting of special events and gives good experimental results.

This method decreases training time considerably. The simulation results have shown the potential of proposed network to tackle highly complex load forecasting problem.

The network is capable of handling the inherent complexity of load forecasting problem.

As an example of another research which uses RWN in their work we have [62].

2.4.5 Neural network-fuzzy methods

Kim et al. in [27] proposed a new short-term load forecasting method for special days in anomalous load conditions. The proposed method uses a hybrid approach of ANN based technique and fuzzy inference method to forecast the hourly loads of special days. In this method, special days are classified into five different day-types. Five ANN models for each day-type are used to forecast the scaled load curves of special days, and two fuzzy inference models are used to forecast the maximum and the minimum loads of special days. Finally, the results of the ANN and the fuzzy inference models are combined to forecast the 24 hourly loads of special days. The proposed method was tested with actual load data of special days for the years of 1996-1997. The test results showed very accurate forecasting with the average percentage relative error of 1.78%.

The fuzzy theory is actively utilized to reduce the uncertainty and the nonlinear property which are latent to the problem of load forecasting on special days [10, 8].

The fuzzy inference method minimizes model errors and the number of the membership functions to grasp nonlinear behavior of power system loads.

The concept of fuzzy regression analysis was introduced by Tanaka et al. [63], where a linear programming (LP) - based method with symmetric triangular fuzzy parameters was proposed. Fuzzy data analysis, regarded as a nonstatistical procedure for probablilistic systems was reported by Tanaka et al. [64]. The fuzzy regression approach showed usefulness to problems of load forecasting and load estimation in power distribution systems [65, 66].

In the paper [10], a new fuzzy linear regression method for the short-term load forecasting of the holidays was proposed. An improved Tanaka's fuzzy regression model [63], and the fuzzy regression approach [65] by introducing fuzzy input-output data using shape-preserving fuzzy arithmetic operations. Coefficients and both input and output data are considered as fuzzy numbers. The new fuzzy regression model improves the prediction accuracy for the short-term load forecasting of the holidays falling on any type of day. The maximum average percentage error obtained was 3.57% in the short-term 24 hourly loads forecasting of the holidays for the years of 1996–1997.

Other researchers like [29, 67, 68, 69, 70, 71, 72, 73, 74] use also the fuzzy approach in its works.

2.5 A *densification trick* using ITL Mean Shift to allow demand forecasting in special days

In the recent paper [4], Sumaili, Miranda et al. proposed a new method to resolve the problem of the lack of historical data in special days. They were inspired by the results of the Information Theoretical Learning Mean Shift algorithm aplied in a process denoted *densification trick* successfully applied in a problem of incipient fault diagnosis in power transformers [75], where scarce data on failures existed.

Thus, the ITLMS algorithm was used to populate, with virtual data, a scarce set related to daily energy consumption in special days. This allows the proper training of neuronal networks with the virtual data, reserving all the scarce real data for validation purposes. The networks are then used to predict consumption in special days. An example with real data from a Brazilian distribution utility was used in order to illustrate this technique.

The remarkable accuracy achieved in forecasting for holidays confirmed the correctness of this new approach. With the division of the data set by the five work days of the week (from Monday to Friday) was obtained the following forecasting indicators: the NMAE varied from 1.85% to 3.92%, while the variation range of the std was from 1.66% to 2.50%.

It is important to note that this results was obtained using a simple neural network for each cluster cluster of special days. More sophisticated arrangements of neural networks are likely to allow further improvement with a narrower accuracy.

2.6 Conclusion

Through the analysis of recent research in the area of load forecasting on special days, it was verified that good results have been achieved and visible improvements in new methodologies was achieved compared with the traditional methods.

It has been proven that the methods based on ANN are good approximators with the capability of modeling every nonlinear system.

However, conventional ANN based short-term load forecasting techniques have limitations in their use on special days. Therefore, some researchers suggest new techniques with integrated models that reduce the uncertainty and the nonlinear property of the load, in this way it was possible to minimize the forecast errors.

Despite the success of these methods none of them has been able to solve the problem of the lack of historical information on special days. So this thesis has as main purpose the resolution of this problem based in the recent work of Sumaili, Miranda et al. [4]. The method implemented will be described below in more detail.

Chapter 3

Tools

In this chapter the used tools in this thesis work will be presented. These main tools include the Information Theoretic Learning Mean Shift (ITLMN) algorithm, Autoassociative Neural Networks (AANN) also known as Autoencoders and Metaheuristics methods, with focus in Evolutionary Computation (EC) algorithms (Evolutionary Algorithm (EA), Particle Swarm Optimization (PSO) and Evolutionary Particle Swarm Optimization (EPSO)).

The research where the autoencoder structure to implement the new method to load forecasting on special days was inspired, will also be addressed.

Therefore, the following chapter seeks to inform more about these tools, their advantages and how they are currently applied.

3.1 Information Theoretic Learning Mean Shift

The Information Theoretic Learning Mean Shift (ITLMS) algorithm was introduced by Rao, Principe and Martins [76, 1] as a means to capture the dominant structures in the data set, as embedded in its estimated probability density function (pdf) [75].

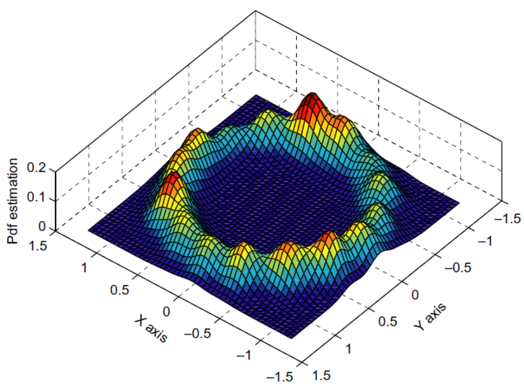


Figure 3.1: pdf estimated, from [1].

In this subchapter this algorithm, as well as its potentialities will be presented.

The Mean Shift algorithm was firstly proposed by Fukunaga and Hostler in 1975 [77]. In this paper they showed that this algorithm is a steepest descent technique where the points of a new dataset are moving in each iteration towards the modes of the original dataset.

Considering a dataset $X_0 = (X_i)_{i=1}^{N} \; \varepsilon \; \mathbf{R}^D$, using the nonparametric method of parzen window technique [78] and a gaussian kernel given by $G(t) = e^{-\frac{t}{2}}$ with bandwidth $\sigma > 0$, the pdf can be estimated by:

$$p(x,\sigma) = \frac{1}{N}\sum_{i=1}^{N} G\left(\left\|\frac{x-x_i}{\sigma}\right\|^2\right) \tag{3.1}$$

The objective of this algorithm is to find the modes of the dataset where $\nabla p(x) = 0$. With that in mind, the iterative stationary point equation is:

$$m(x) = \frac{\sum_{i=1}^{N} G\left(\left\|\frac{x-x_i}{\sigma}\right\|^2\right) \cdot x_i}{\sum_{i=1}^{N} G\left(\left\|\frac{x-x_i}{\sigma}\right\|^2\right)} \tag{3.2}$$

The difference $m(x) - x$ is known as mean shift.

In literature, this first algorithm is known as Gaussian Blurring Mean Shift (GBMS) indicating the successive blurring of the dataset towards its respective modes due the actual solution being a single point that minimizes the overall entropy of the data set.

In spite of this important development, the Mean Shift idea was forgotten until 1995, when Cheng [79] introduced a slight change in the algorithm. While in Fukunaga's algorithm the original dataset is forgotten after the first iteration, $X^{(0)} = X_0$, the Cheng's algorithm keeps this dataset in memory. This initial dataset is used in every iteration to be compared with the new dataset Y.

However Y is initialized the same way, $Y^{(0)} = X_0$. This also introduces a small change in the iterative equation:

$$m(x) = \frac{\sum_{i=1}^{N} G\left(\left\|\frac{x-x_{0i}}{\sigma}\right\|^2\right) \cdot x_{0i}}{\sum_{i=1}^{N} G\left(\left\|\frac{x-x_{0i}}{\sigma}\right\|^2\right)} \tag{3.3}$$

In literature, this algorithm changed is known as Gaussian Mean Shift (GMS) Algorithm.

Mean Shift algorithms have been shown a very versatile and robust tool in feature space analysis [80] and is often used in image segmentation [81, 82], denoising, tracking objects [83] and several other computer vision tasks [84, 85].

Recently, in 2006, Rao, Principe and Martins [76, 1] introduced a new formulation of mean shift known as Information Theoretic Learning Mean Shift (ITLMS) and showed that GBMS and GMS are special cases of this one.

The idea in this algorithm was to create a cost function that minimizes the cross entropy of the data while the Cauchy-Schwartz distance is kept at a given value.

Knowing that a gaussian kernel (with bandwidth $\sigma > 0$) is given by:

$$G_\sigma = e^{\frac{-x^2}{2 \cdot \sigma^2}} \tag{3.4}$$

an estimation of a pdf, using the parzen window technique [78], is:

$$p(X) = \frac{1}{N} \sum_{i=1}^{N} G_\sigma (x - x_i) \tag{3.5}$$

Renyi's quadratic entropy [86] for a pdf can be calculated using:

$$H(X) = -\log \int_{-\infty}^{+\infty} p^2(x)\, dx \tag{3.6}$$

Therefore, replacing 3.5 into 3.6,

$$H(X) = -\log V(X) \tag{3.7}$$

with

$$V(X) = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} G_{\sigma'}(x_i - x_j) \tag{3.8}$$

where $\sigma' = \sqrt{2}\sigma$. $V(x)$ is known as the *information potential* of the pdf $p(X)$. The derivative of this expression with respect to a single point x_i gives a quantity denoted *information force* exerted by all data particles on x_i.

To measure the *cross entropy* between two pdf, one has

$$H(X, X_0) = -\log V(X, X_0) \tag{3.9}$$

with

$$V(X, X_0) = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} G_{\sigma'}(x_i - x_{0j}) \tag{3.10}$$

The Cauchy-Schwartz distance between two pdfs (p and q) can be calculated using:

$$D_{CS}(X, X_0) = \log \left(\frac{(\int p^2(x)dx) \cdot (\int q^2(x)dx)}{(} \int p(x) \cdot q(x)dx)^2 \right) \tag{3.11}$$

$$D_{CS}(X, X_0) = -[H(X) + H(X_0) - 2H(X, X_0)] \tag{3.12}$$

The ITLMS algorithm aims at finding data sets X that capture structural information from a set X_0. This is achieved by a double criteria optimization, minimizing the entropy of X while keeping the Cauchy–Schwartz distance at some value k. An unconstrained optimization formulation, under a parameter λ (Lagrange multiplier) that represents the tradeoff between the two objectives is given by

$$J(X) \quad = \quad min\, H(X) \;+\; \lambda \cdot [D_{CS}(X, X_0) - k] \tag{3.13}$$

Differentiating $J(X)$ with respect to each x_i gives an algorithmic rule that allows the transformation of X_0 into another set at iteration $t+1$, making use of the information contained in the pdf of X at iteration t, estimated by 3.5:

$$x_i^{t+1} \quad = \quad \frac{c_1 \cdot S_1 \;+\; c_2 \cdot S_2}{c_1 \cdot S_3 \;+\; c_2 \cdot S_4} \tag{3.14}$$

where

$$c_1 \quad = \quad \frac{1-\lambda}{V(X)}, \qquad c_2 \quad = \quad \frac{1-\lambda}{V(X, X_0)} \tag{3.15}$$

and

$$S_1 \quad = \quad \sum_{j=1}^{N} G_\sigma \left(\frac{\|x_i^t \;-\; x_j^t\|^2}{\sigma'} \right) \;\times\; x_j^t \tag{3.16}$$

$$S_2 \quad = \quad \sum_{j=1}^{N} G_\sigma \left(\frac{\|x_i^t \;-\; x_{0j}^t\|^2}{\sigma'} \right) \;\times\; x_{0j} \tag{3.17}$$

$$S_3 \quad = \quad \sum_{j=1}^{N} G_\sigma \left(\frac{\|x_i^t \;-\; x_j^t\|^2}{\sigma'} \right) \tag{3.18}$$

$$S_4 \quad = \quad \sum_{j=1}^{N} G_\sigma \left(\frac{\|x_i^t \;-\; x_{0j}^t\|^2}{\sigma'} \right) \tag{3.19}$$

As shown in [76], adjusting the λ parameter changes the data properties sought by the algorithm:

$\lambda = \mathbf{0}$ – the algorithm minimizes the data entropy, returning a single point. This is the GBMS algorithm;

$\lambda = \mathbf{1}$ – the algorithm is a mode seeking method. The particles converge to the modes of the pdf $p(X)$, the same as GMS;

$\lambda > \mathbf{1}$ – the principal curve of the data is returned ($1 < \lambda < 2$). A higher value of λ makes the algorithm seek to represent all the characteristics of the pdf.

Each generation of points x_i^t describe a pdf $p(X^t)$ that retains information from $p(X_0)$. Each point x_i^t along the iterations t describes a path from x_{i0} toward a mode of the pdf $p(X_0)$, or to a principal curve of the data cluster, or to a region of higher density, depending on the value of λ adopted. By path, one means a succession of points $X^0, X^1, ..., X^t, ...$ that may be driven toward or away from the mode, depending on allowing points to follow the direction of the information force ($\partial V / \partial X$ as in 3.8) or the reverse direction.

The set $X_V = X^1 \cup X^2 ... \cup X^t$ is the set of virtual data generated by the ITLMS algorithm. It forms a dense cluster that shares properties with the original X_0.

This use of X_V is called the *densification trick*.

This property was successfully applied in a problem of incipient fault diagnosis in power transformers [75] by Miranda et al., where scarce data on failures existed.

Therefore, this *densification trick* becomes especially useful when data is scarce or often insufficient to a neural network training practice.

The insufficient number of samples is a difficulty present in many works reported. In particular, the solidity of models whose validation rests on such a low number of test samples may be questioned.

With the use of the ITLMS, the training set may be composed of only virtual points, keeping the totality of the real data to be used in the testing phase. This largely increases the robustness of the testing procedure and the confidence in the results it will provide.

The *densification trick* using ITLMS demonstrates to be a powerful tool to resolve the problem of the typical lack of historical data on special days, as demonstrated in [4] by Sumaili, Miranda et al.. Its application in the construction of a neural network system for the 1 day-ahead prediction of electric energy consumption in special days was suggested, for a Brazilian distribution utility.

In chapter 4 the results of the densification of data sets which will be applied as pratical example in this thesis will be presented.

3.2 Autoassociative Neural Networks

For a better understanding of which is an Autoassociative Neural Networks (AANN), the next general contextualization of Artificial Neural Networks will be provided.

Artificial Neuronal Networks, or just Neuronal Networks (NN) are machines designed the way human brain performs/learns a particular task or function of interest [2]. NN provide a principled framework for learning linear and non-linear mappings from an input to an output space, corresponds to a connectionist paradigm of information processing, including a massive paralel process of numerical computacions [6], through a process of learning. The basic processing element of a NN is the neuron [87]. Neurons are composed of several inputs, one output and an activation function which executes the internal processing, transforming the inputs into the output. Usually, neurons are organized in layers with unidirectional links always in a forward direction, from the input to output of the NN(feedforward networks).

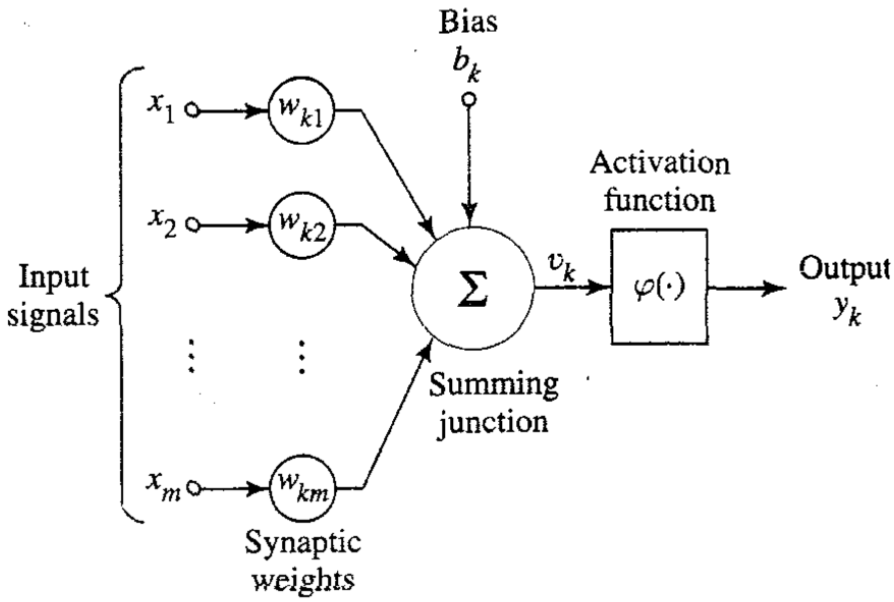


Figure 3.2: Nonlinear Model of a Neuron, from [2].

Connections between neurons are associated with synaptic weights w_{kj}, such that a signal emitted by a neuron is multiplied by the weight of the conection before entering a next neuron [6]. This process is schematized in figure 3.2 and the equations demonstrated are:

The weighted sum of inputs x_j:

$$u_k = \sum_{j=1}^{m} w_{kj}.x_j \tag{3.20}$$

The summing junction of the bias b_k to the u_k:

$$v_k = u_k + b_k \tag{3.21}$$

Finally, the output y_k is the result of v_k through activation function $\varphi(\cdot)$.

$$y_k = \varphi(v_k) \tag{3.22}$$

Autoassociative Neural Networks (AANN), also known as autoencoders, are feedforward neural networks with a middle hidden layer that intends to reconstruct the output equal the input.

Thereby, the size of the output layer is always the same as the size of the input layer. The simplest autoencoder architecture has only one middle hidden layer, once the use of more hidden layers makes the training tedious and furthermore, it will also not give good results [88]. The optimal number of the hidden neurons, thought dependent on the type of application, must be smaller than that of the input and output. In the figure 3.3, a typical diagram of an autoencoder is shown.

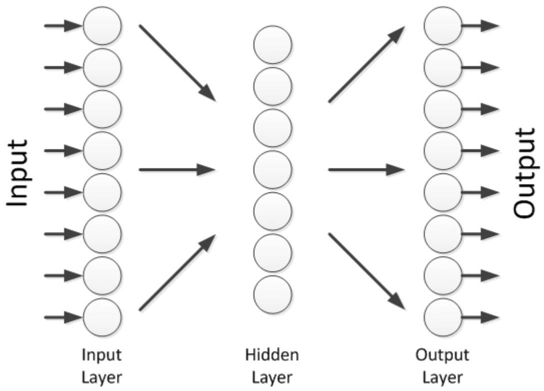


Figure 3.3: The structure of a eigth-input, eight-output autoencoder

The autoencoders perform two main operations: a forward compressing operation transforming from data space to code space at the hidden layer called "encoding", and reverse transformation from code space to data space at the output layer called "decoding".

If linear activation functions are used, autoencoder will be performing similar to Principal component analysis (PCA) method [89, 90] that is, reduce the dimensionality of data. With nonlinear activation functions, autoencoders chart the input space on a nonlinear manifold in such a way that an approximate reconstruction is possible with less error [91]. Plus, PCA does not easily show how to do the inverse reconstruction, which is straightforward with autoencoders [92].

The goal is to train the network such that the composed operation is as close as possible to the identity mapping. By defining a network structure where inputs and outputs are tied to training samples, appropriate network parameters (weights and biases) can be trained by on criterion optimization, the classical function adopted is the minimization of the Mean Square Error (MSE) between the input and the outputs.

If X is the input vector and Y the output vector, then for N samples:

$$MSE: \quad min\ \varepsilon \quad = \quad min\ \frac{1}{N}\sum_{k=1}^{N} \|X_k - Y_k\|^2 \tag{3.23}$$

A good interpretation of the MSE criterion is that it represents the minimization of the variance of the pdf of error distribution. However, this criterion is optimal only if this distribution is

Gaussian, which may be questionable in many applications where a non-parametric method may achieve a better result [75].

The reconstructions can be learnt via some algorithms like gradient descent or the backpropagation.

To train a neural network, two independent data sets are needed: one to train the network and another to validate its results. While the training set is used to adjust the connection weights, the validation set is used to verify if the network is generalising in a proper way. Generalising is the neural network ability to recognize points with the same properties of the ones in the training set but didn't belong to it.

Therefore, one interesting property of autoencoders is that when the network is properly trained, it may be used as a recognition machine. If a new input vector provides different characteristics from the global pattern of the data used for training, the error between the output and input tends to be high, since the result does not match the input. This is extremely useful in pattern recognition tasks as an approach to missing data in database estimation [93, 94, 95, 96] and it is very important to the work done in this thesis.

Autoencoders are often used to compress data like images [97, 98, 99], or for instance, face images could be identified and clustered according to sex, distinguished from non-faces [100], etc. Another application is the reconstruction of missing data, used in applications to missing sensor restoration [101, 102, 103], reconstructing missing data in state estimation [92], diagnosing faults in power transformers [75] and several other applications [104, 105, 106, 107].

3.3 Missing Data Estimation Using Autoencoders

A great deal of research has recently been done to discover new ways of estimating the missing values in databases. Among others, Abdella and Marwala [93] and Mohamed and Marwala [96] used neural networks together with Genetic Algorithms (GA) to approximate missing data. Qiao et al. [103] used neural networks and Particle Swam Optimisation (PSO) to keep track of the dynamics of a power plant in the presence of missing data. Dhlamini et al. [105] have used Evolutionary computing in condition monitoring of high voltage (HV) bushings in the presence of missing data. Miranda et al. [92] used neural networks together with Evolutionary Particle Swam Optimisation (EPSO) to solve the problem of recomposing missing information at the SCADA of energy/distribution management systems (EMS/DMS). In their study, auto-associative neural networks were used together with GA, PSO or EPSO to predict the missing data and also to optimise the prediction.

The optimization algorithms (GA, PSO or EPSO) are used to estimate the missing values by optimizing an objective function. The complete vector combining the estimated and the observed values is fed into the auto-encoder as input and as shown in figure 3.4. Symbols X_k and X_u represent the known variables and the unknown or missing variables, respectively. The combination of X_k and X_u represent the full input space.

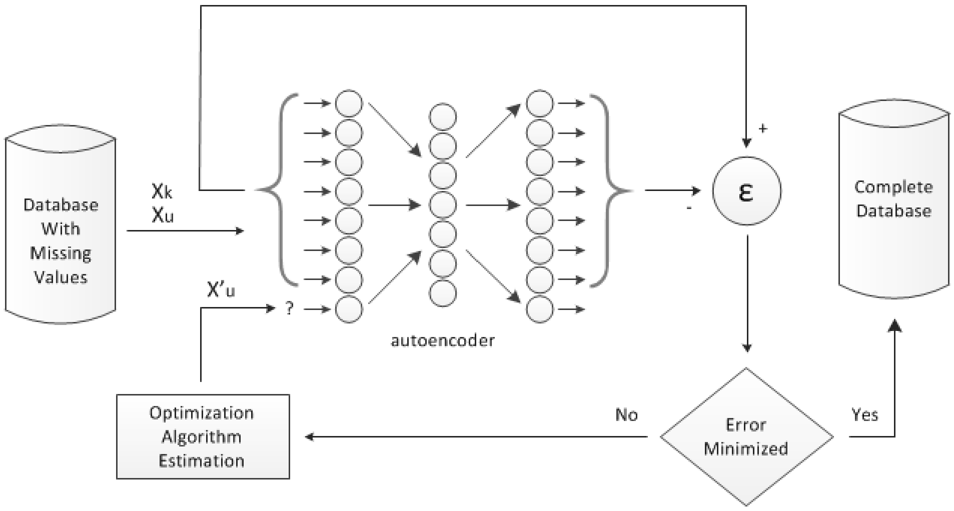


Figure 3.4: Autoencoder and optimization algorithm based missing data estimator - Constrained Search model

This method uses an autoencoder, so it will be expected that through a well chosen architecture, the input and the output are very similar. This is, however, only feasible on a dataset similar to the problem space, as outlined above. In other words, the autoencoder is trained to minimize the difference between its input and output vectors only possible for an input vector with satisfactorily similar properties to those of the input vectors upon which it was trained. This procedure is what allows the autoencoder to estimate the missing data. The difference between the target and the actual output is used as error. In [93] Abdella and Marwala used the square of the error to create a function which has a global optimum where the difference between the target and the output is zero. By squaring this error, the zero-crossing of the linear error becomes the minimum error of the quadratic error.

This leads to the following equation:

$$\varepsilon = \left(\left\{\begin{pmatrix} X_k \\ X_u \end{pmatrix}\right\} - \mathrm{f}\left(\overrightarrow{W}, \left\{\begin{pmatrix} X_k \\ X_u \end{pmatrix}\right\}\right)\right)^2 \tag{3.24}$$

where X and $\overrightarrow{W}$ are input and weight vectors whereas X_k and X_u represent the known and unknown input variables, respectively. This equation is used as the objective function that is minimized using GA. More details about the base method can be found in [93]. On the other hand, the publications [102] and [92] describe some useful properties of autoencoders in restoring missing values. After the autoencoder is adequately trained, basic approaches such as unconstrained search and constrained search can be considered, the study of these methods have the purpose to find a better technique for discovering the true point of convergence.

Constrained Search model controls the error convergence in all the input-output data of the autoencoder.

$$min\ \varepsilon \quad = \quad min\ \left(\left\{\begin{pmatrix} X_k \\ X_u \end{pmatrix}\right\} - \mathrm{f}\left(\overrightarrow{W}, \left\{\begin{pmatrix} X_k \\ X_u \end{pmatrix}\right\}\right)\right)^2 \tag{3.25}$$

where X and $\overrightarrow{W}$, as in the function 3.24, are input and weight vectors, X_k and X_u represent the known and unknown input variables, respectively. In the figure 3.4 this approach can be seen.

On the other hand, Unconstrained Search model controls the convergence by the optimization algorithm in order to minimize the input-output error only on the missing data. The following equation can be regarded:

$$min\ \varepsilon \quad = \quad min\ \left(\{X_u\} - \mathrm{f}\left(\overrightarrow{W}, \{X_u\}\right)\right)^2 \tag{3.26}$$

where X_u and $\overrightarrow{W}$ are unknown input variables and weight vectors. In the following figure this model can be seen.

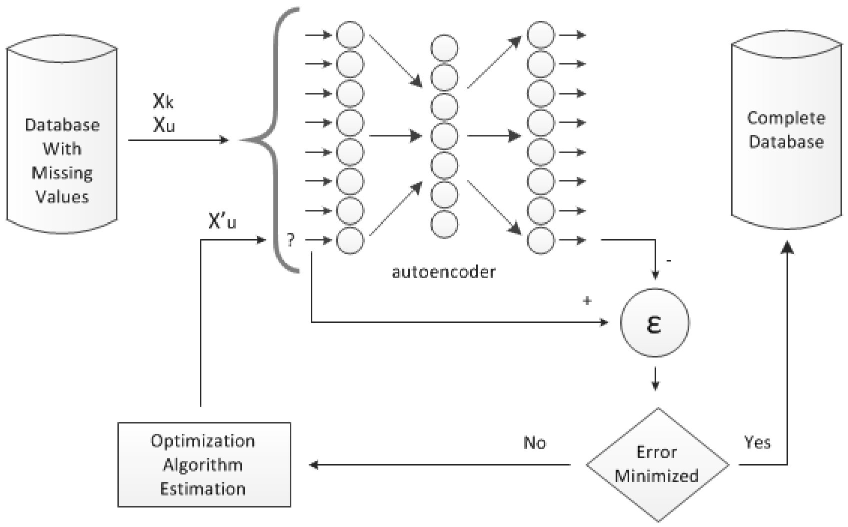


Figure 3.5: Autoencoder and optimization algorithm based missing data estimator - Unconstrained Search model

The Autoencoder training will be preformed by the MATLAB© Neural Network Toolbox™ 7 software. For more details consult the User's Guide [5].

3.4 Metaheuristic Methods

Metaheuristic methods employ smart strategies for searching in the solution space, the best solution in a quick and efficient manner. Metaheuristics can be organized as it is shown in the next figure.

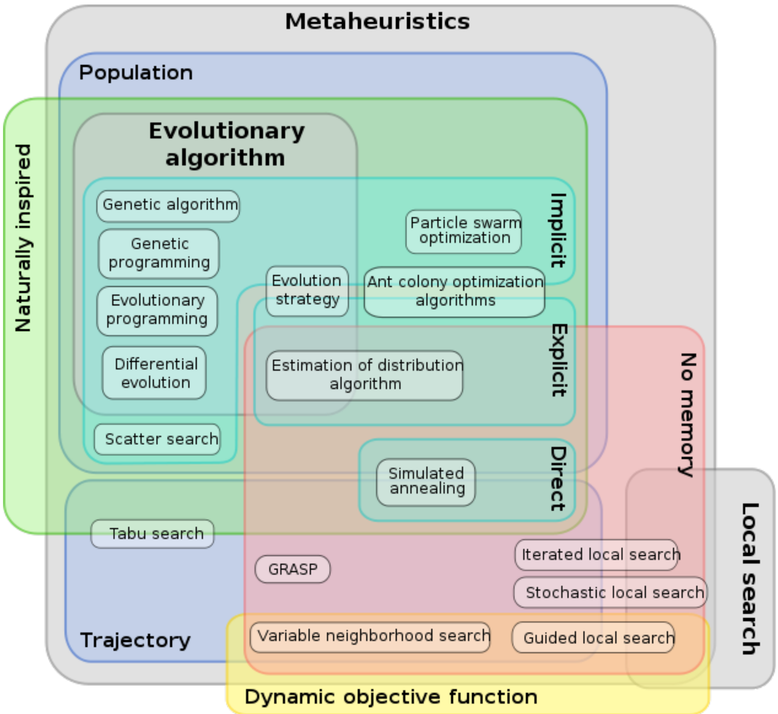


Figure 3.6: Metaheuristics classification.

The evolutionary computation is based on the evolutionary mechanisms found in nature. These mechanisms are directly related with Darwin's Theory of Evolution, where he states that life on Earth is the result of a selection process, done by the environment in which only the fittest and adapted ones possess chances of surviving and consequently reproduce itself [108].

The goal in this optimization is to find the best possible solution or solutions to a problem, whatever the nature of its variables, with respect to one or more criteria. To do that a population or set of possible solutions (individuals) to the problem is formed. Each individual is evaluated by an optimization function, and the best are selected to reproduce. Then new individuals from the selected will be produced and the new generation is formed. This new generation is assessed and the individuals with the worst performance are eliminated prevailing the best individuals and a new phase of reproduction originates the subsequent generation. This process is repeated generation after generation, and the population should keep improving itself with the individuals with better assessment until a certain stop criterion is satisfied. The best individual found in this process is taken as the solution of the optimization problem.

In this way, the various methods of evolutionary computation can be distinguished by the following factors:

- the form of representation (chromosome) of a solution or individual;
- the form of decoding the chromosomes;
- the form of making the selection;
- the form of making the reproduction (or generation of new individuals).

The evolutionary computation, for historical reasons, has been divided for several years in Evolution Strategies (ES) and Evolutionary Programming (EP) but nowadays it does not make sense to divide these two variants once they correspond to phenotype methods.

ES/EP diverge of Genetic Algorithms GA (genotype methods) in the way they represent the alternatives, solutions or individuals of the population. GA are based on the genetic discrete representation of each individual to generate new individuals with better possibilities to survive while ES/EP are based on the direct representation of the solutions and are only based in the own problem variables without passing through any intermediate algorithm of encoding / decoding.

In evolutionary algorithms (EA) the searching mechanism is constituted by the action of mutation and recombination, where the concurrence of these two operators proposes new points in space departing from previous locations, which are then subject to evaluation and selection. It is the presence of a selection operator that distinguishes evolutionary algorithms from other meta-heuristics.

As is described by Miranda et al. [109] Particle Swarm Optimization (PSO) algorithms have no selection operator, but a specific movement rule is adopted that defines how a new particle is created departing from its history and from information from the swarm. Under controlled circumstances, this can drive the swarm to the optimal solution of a problem. The PSO movement rule has dynamic characteristics that drive the optimization process towards the optimum without requiring selection. Eberhart and Kennedy developed PSO, inspired in the analogy of swarms of insects, flocks of birds, schools of fish or other groups, in which the behavior of each individual is simultaneously influenced by own factors and factors (social) that result from the behavior of others [110]. This is a concept far from the classical paradigm of EA.

Therefore, in PSO the particles move under the action of three influences (vectors) that complement each other and are called inertia, memory and cooperation. The first vector pushes the particle in a direction identical to that that had been following. The second vector attracts the particle toward the best position occupied by the particles during their life. The third vector attracts the particle towards the best point in space so far discovered by the swarm.

With PSO, unlike the EA, there is no competition between particles or auto-adaptation of their characteristics. From the beginning their promoters realized the need to introduce controls on the behavior of swarms, or to enhance the efficiency of search you want to avoid divergence of the swarm. These controls have been, in most cases, externally applied, based on empirical recipes, and have begun testing ideas of self-adaptation [108].

This idea of self-adaptation inspired a new concept of evolutionary algorithms the Evolutionary Particle Swarm Optimization (EPSO) algorithms [111] based in evolution strategies. EPSO is an evolutionary swarm optimization algorithm meta-heuristic that combines the concepts of evolutionary strategies, which is a characteristic of EA, with particle swarm optimization of PSO.

EPSO seeks to give an adaptive character of the particles swarm algorithms, being then predictable, that this method in many applications can reach better results than the classic PSO [3, 108].

Then these two algorithms, PSO and EPSO, will be described in more detail.

3.4.1 Particle Swarm Optimization (PSO)

The PSO algorithm has been presented as illustrating the movement of a set of particles exploring the space of solutions or decisions of n dimension according to the number of problem variables. The simple model of PSO is described below as Miranda relates in [108].

Each particle corresponds to an alternative solution for a given optimization problem. Given a population of n particles, each particle i has the following composition:

- A position vector X_i;
- A velocity vector V_i;
- A memory vector b_i of the best position found during his lifetime;
- A value of the objective function relative of the current position X_i;
- A value of the objective function relative of the best position found by the particle b_i.

At any given instant t (corresponding to a given iteration) i a particle changes its position X_i to X_i^{new} according to the following movement rule:

$$X_i^{new} = X_i + V_i^{new} \tag{3.27}$$

where V_i^{new} is the new velocity of the particle i, i.e., the vector representing the change of position of the particle i and is given by

$$V_i^{new} = V_i + Rnd \cdot Wm_i \cdot (b_i - X_i) + Rnd \cdot Wc_i \cdot (b_g - X_i) \tag{3.28}$$

where

$\mathbf{Wm_i}$ weight conditioning the memory term;

$\mathbf{Wc_i}$ weight conditioning the cooperation term;

$\mathbf{b_i}$ best position found by the particle in its past life;

$\mathbf{b_g}$ best position found by the swarm of particles in their past life;

Rnd random numbers sampled from a uniform distribution in [0,1].

The following figure illustrates this concept.

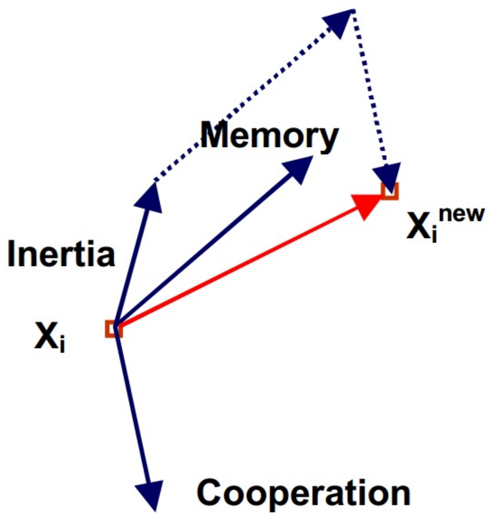


Figure 3.7: Ilustrating the movement of a particle i in PSO, influenced by the three terms: Inertia, Memory and Cooperation [3].

The weights affecting the various terms are affected in each iteration by multiplying random numbers, which causes a disturbance in the trajectory of each particle which has been shown to be beneficial for space exploration and discovery of the optimal solution.

The weights in this simple model are defined initially and externally. This raises a tuning problem of these weights in order to reach the convergence. In fact, this is the main disadvantage of PSO algorithm, it is not self-adaptative.

In order to achieve better results we can apply mechanisms in the movement rule that, although none of them solve the problem of the lack of self-adaptivity, it can bring improvements to the PSO algorithm [3].

There exists then two principal mechanisms, one of them can be described in the following manner:

$$V_i^{new} = Dec(t) \cdot Wi_i \cdot V_i + Rnd \cdot Wm_i \cdot (b_i - X_i) + Rnd \cdot Wc_i \cdot (b_g - X_i) \quad (3.29)$$

I.e. apply in the inertia term a $Dec(t)$ function whose value decreasing with the progress of iterations, reducing progressively the importance of this term [112], and also apply a new weight Wi_i.

The other mechanism (proposed by Maurice Clerc [113]) consists in the multiplication of the movement rule by a constriction factor K. This factor consists of a diagonal matrix of constriction factors of dimension k.

$$K_k = \frac{2}{|2 - W_k - \sqrt{W_k^2 - 4 \cdot W_k}|}, \quad W_k = Wm_K + Wc_k,\ W_k > 4 \tag{3.30}$$

3.4.2 Evolutionary Particle Swarm Optimization (EPSO)

As described above the EPSO algorithm can be seen as a hybrid method of ES/EP and PSO techniques. As an ES, an EPSO algorithm may be described (as Miranda in [3]) by the following general scheme:

Replication each particle is replicated n times;

each particle has its strategic parameters mutated;

Reproduction each mutated particle generates an offspring through recombination, according to the particle movement rule, described below;

Evaluation each offspring has its fitness evaluated;

Selection by stochastic tournament or other selection procedure, the best particles survive to form a new generation, composed of a selected descendant from every individual in the previous generation.

The EPSO reproduction rule can be described by the following expression where given a particle X_i, a new particle X_i^{new} will be:

$$X_i^{new} = X_i + V_i^{new} \tag{3.31}$$

The movement rule of the EPSO is given by

$$V_i^{new} = Wi_i^* \cdot V_i + Wm_i^* \cdot (b_i - X_i) + Wc_i^* \cdot (b_g^* - X_i) \cdot P \tag{3.32}$$

where

$\mathbf{Wi_i}$ weight conditioning the inertia term;

$\mathbf{Wm_i}$ weight conditioning the memory term;

$\mathbf{Wc_i}$ weight conditioning the cooperation term;

$\mathbf{b_i}$ best position found by the particle in its past life;

$\mathbf{b_g}$ best position found by the swarm of particles in their past life;

P communication factor, assumes binary variables of value 1 with probability p and value 0 with probability (1-p); the p value, set as an external parameter, controls the passage of information within the swarm and is considered 1 in classical formulations.

EPSO algorithms include the adoption of a stochastic star communication topology, instead of the deterministic scheme usually adopted in PSO. This has the advantage of sharing the knowledge of each particle's knowledge of the global best position, controlled by a communication probability P, which is self-adaptive throughout the algorithm run and also externally defined. The effect produced by the adoption of a stochastic star communication topology is that a particle will ignore the global best on some iterations and include it in other iterations. This not only allows more local search by each particle, but also allows the elimination of disturbing noise, by allowing the dynamics of particle movement to be more stable and avoiding premature convergence [109].

The symbol $*$ indicates that these parameters will undergo evolution under a mutation process.

The difference for the particle swarm PSO is that the evolution does not only occur in the behavior of particles, but also on the weights that affect the movement of these in the search space. One of the main features is that it is a self-adaptive method, that is, automatically adjusts the swarm behavior in order to enhance efficiency in the search and on the other hand, prevent the divergence of the swarm.

This characteristic lies in the fact that at a given instant, there is a particle which has the best position in the search space, and the population of particles have to move in this direction. In addition, each particle is also attracted to its previous best position.

This process of the EPSO is illustrated in the following figure 3.8.

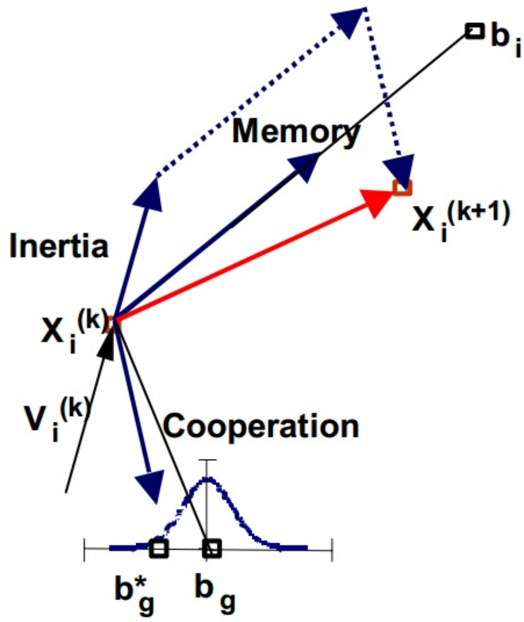


Figure 3.8: Illustration of EPSO particle reproduction: a particle X_i generates an offspring at a location commanded by the movement rule [3].

The approximate basic mutation rule for the strategic parameters is the following:

$$Wk_i^* \quad = \quad Wk_i \cdot [\, 1 + \tau \cdot N(0,1) \,] \tag{3.33}$$

where

$\mathbf{N(0,1)}$ random variable with Gaussian distribution (0 mean and variance 1);

τ learning parameter, fixed externally, controlling the amplitude of the mutations – smaller values of τ lead to higher probability of having values close to 1.

As for the global best b_g, it is randomly disturbed to give

$$b_g^* \quad = \quad b_g + Wb_i^* \cdot N(0,1) \tag{3.34}$$

where wb_i is the strategic weight parameter associated with particle i. It controls the size of the interval of bg where it is more likely to find the real global best solution. This weight wb_i is mutated (denoted by *) according to the general mutation rule.

In several papers it is possible to verify the advantages of this optimization tool in electrical power applications [111, 114, 115, 116, 117, 118, 119, 120].

Chapter 4

Data Treatment

The historical data which were taken into account in this thesis are the same that were applied in [4] by Sumaili, Miranda et al., the real data from a Brazilian distribution utility. The historical data refer to about 10 years of consumption (from January 2002 to September 2012).

In Brazil, public holidays may be legislated at the federal, statewide and municipal levels. Most holidays are observed nationwide, but each state and city may have its own holidays as well.

Apart from the yearly official holidays (listed below), the Constitution of Brazil also establishes that election days are to be considered national holidays as well. General elections are held on the first Sunday of October, in the first round, and on the last Sunday of October, in the second round, of every even year.

Table 4.1: National holidays in Brazil.

Date	**Holiday name**	**Holiday type**
January 1	New Year's Day	Fixed by date
47 days before Easter	Carnival/Shrove Tuesday	Fixed by day (Tuesday)
Day after Carnival	Carnival end (until 14 hrs)	Fixed by day (Wednesday)
Friday before Easter	Good Friday	Fixed by day (Friday)
Computus[1]	Easter Day	Fixed by day (Sunday)
April 21	Tiradentes Day	Fixed by date
May 1	Labour Day	Fixed by date
Thursday after Trinity Sunday[2]	Corpus Christi	Fixed by day (Thursday)
September 7	Independence Day	Fixed by date
October 12	Our Lady of Aparecida	Fixed by date
November 2	All Souls Day	Fixed by date
November 15	Republic Proclamation Day	Fixed by date
December 25	Christmas Day	Fixed by date
December 31	New Year's Eve (from 14 hrs)	Fixed by date

[1]The *Computus* (Latin for "computation") is the calculation of the date of Easter, the first Sunday after the first ecclesiastical full moon (that follows the Northern spring equinox) falling on or after 21 March

[2]Trinity Sunday is the first Sunday after Pentecost. Pentecost is celebrated seven weeks (50 days) after Easter Sunday, hence its name.

Other days can also be considered as special days, like the days preceding the Carnival, the Christmas Eve, the Valentine's Day or even the Fridays or Mondays in extended weekends, among other.

For more details the following website may be consulted: www.timeanddate.com/holidays/brazil/

In this chapter the treatment of the historical data will be described taking into account the following criteria:

- In this study of load forecasting on special days, holidays which occur at Saturdays and Sundays will not be analysed, nor consecutive holidays with frequency inferior to eight days;

- The load forecasting will be based on the daily energy consumption of the days preceding the holiday

- The demand of the same special days are dissimilar each year due to the system load growth/decline trend. If this yearly growth is ignored, the general shapes of same days become similar. Therefore, in this study, like in many others, the load forecasting will be performed based on historical data of holidays with the same behavior (e.g. special days that occur on Wednesdays and which have the same weekly behavior);

- As mentioned earlier, to resolve the problem of the lack of historical data on special days the ITLMS algorithm will be used to make the densification of data set.

4.1 Normalization of Data Set and Its Classification Using ITLMS

As mentioned above, the demand of the same special days are dissimilar each year due to the system load growth/decline trend. If this yearly growth is ignored, the general shapes of same days become similar. Therefore, the load forecasting can be performed based on historical data of holidays which occur at the same day and with the same weekly behavior.

Therefore, the first step to data treatment of the holidays and their previous days is to make a correction in all of them in order to obtain their similarity. The normalization was made with respect to the consumption of the previous week.

The next images illustrates this method.

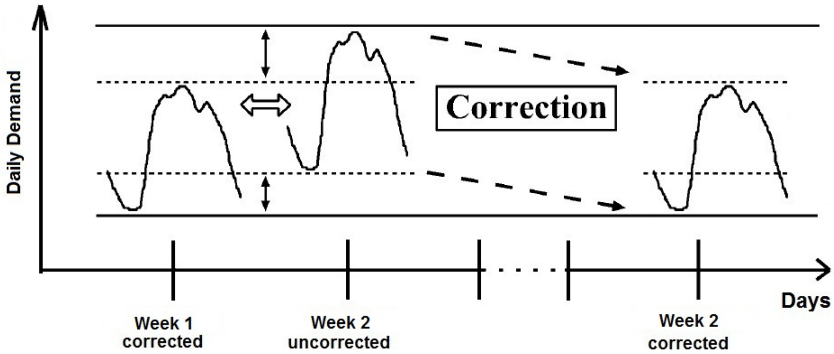
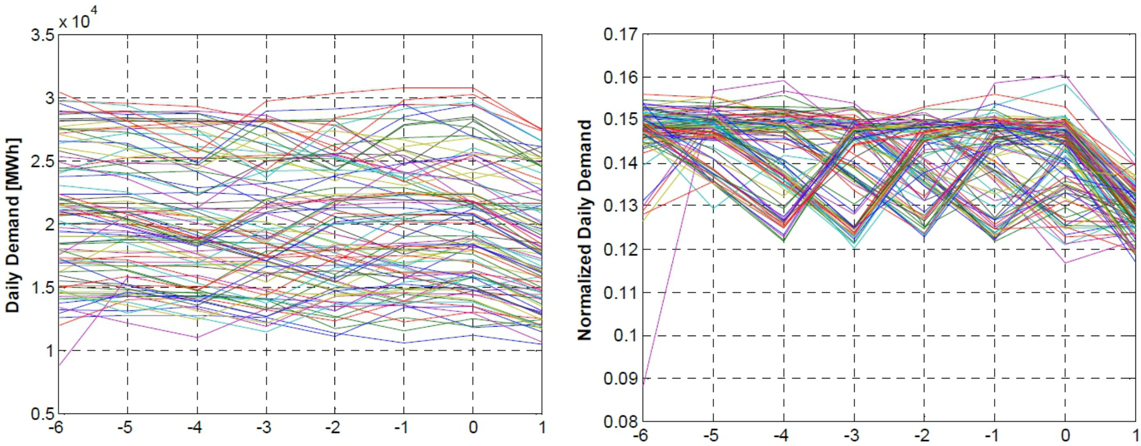


Figure 4.1: Demand correction method.



(a) Set of consumption patterns. (b) Normalized special day patterns.

Figure 4.2: Normalization of the data set, from [4].

It is important to refer which of the special days correspond to the last represented day.

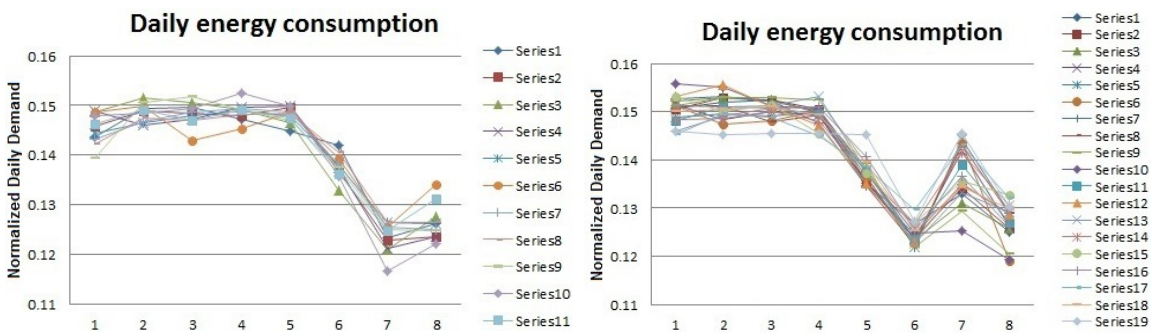
After applying this method, ITLMS algorithm was used to understand the similarity between the patterns shown in figure 4.2b. Therefore, it was possible to identify distinct patterns for special days, and cluster them in similar classes. This way, each special day / holiday was associated to a particular pattern (cluster).

With setting $\lambda = 0.9$ in 3.15 the identification of thirteen different modes was possible. The patterns converging to a common mode were grouped in individual clusters. It was thus possible to form six clusters corresponding to the five days of the week (from Monday to Friday, two clusters on Friday) and others seven groups with the remaining outliers that were not taken into consideration in this study.

In a second approach with $\lambda = 0.1$ in 3.15 ten different modes were identified. Five clusters corresponding to the five days of the week were formed, and other five groups with the remaining outliers that were also not taken into consideration in this study.

The reason for not considering the remaining outliers is because some patterns correspond to a very special cases which should deserve individual analysis. Some of these cases possibly correspond to blackouts, which severely reduced the daily consumption, or also, holidays that do not have a fixed week day distort the observed pattern or even the occurence of two holidays in the same week.

The following figure shows two examples of clusters organized. In the same cluster there are holiday with occurrence on the same day and same weekly behavior.



(a) Monday holidays, cluster with 11 patterns. (b) Tuesday holidays, cluster with 19 patterns.

Figure 4.3: Holidays grouped by the same weekly behavior and which occurred at the same day (Approach 1).

In these two graphs it is possible to observe which different special days (properly standardized) was grouped in the same cluster in order to perform the load forecasting.

The representation of all clusters of patterns can be seen in appendix A.1.

In table 4.2 the number of patterns obtained on database using the properly correction and the ITLMS for classification is shown.

Table 4.2: Database complete description.

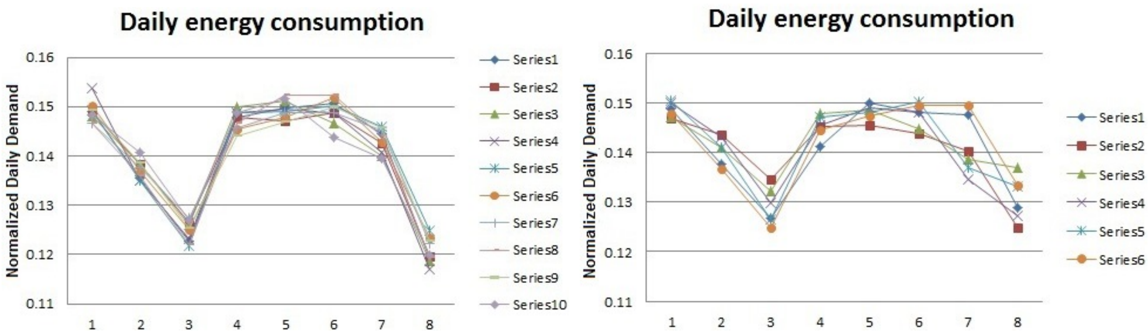
Approach 1 ($\lambda = 0.9$)

Special Day	Real Data
Monday	11
Tuesday	19
Wednesday	14
Thursday	22
Friday	10
Friday	6

Approach 2 ($\lambda = 0.1$)

Special Day	Real Data
Monday	11
Tuesday	22
Wednesday	14
Thursday	22
Friday	19

The reason of the database considering two different groups of special days which occur on Friday in the approach 1 is because these holidays, even falling on same weekday, have a different weekly behavior and the settings given to ITLMS led to consider these days in different clusters (See figure 4.4).



(a) Friday holidays (1), cluster with 10 patterns. (b) Friday holidays (2), cluster with 6 patterns.

Figure 4.4: Holidays grouped by the same weekly behavior and which occurred at the same day.

It is interesting to observe that the lowest value of consumption in each diagram corresponds to the consumption on Sunday, so, it is simple to identify each diagram in association to each day of the week. The phenomenon of "extended weekend" is easily detected in the Tuesday cluster where the consumption on Monday is on average smaller that on the other working days.

4.2 Densification of Data Set

As stated before in section 3.1, the *densification trick* using ITLMS algorithm will be applied as a way of resolving the problem of lack of historical data, insufficient to an AANN training practice.

It is thus intended that the training set is composed of only virtual points, keeping the totality of the real data to be used in the validation phase. This largely increases the robustness of the validation procedure and the confidence in the results it will provide.

The ITLMS algorithm for each cluster was run in order to create a dense cluster of virtual data.

In table 4.3 the full description of the database obtained is shown. The different values of virtual data can be justified with the number of the original real data and the number of iterations needed by the mean shift algorithm to converge to a single mode.

In this study the convergence was reached after 21 iterations, generating 21 virtual patterns for each original data point in approach 1. In approach 2 the convergence was reached after 59 iterations, generating 59 virtual patterns for each original data point. I.e., the total number of virtual patterns in each cluster is the number of original real points times the number of iterations performed.

Table 4.3: Database complete description.

Approach 1 ($\lambda = 0.9$)

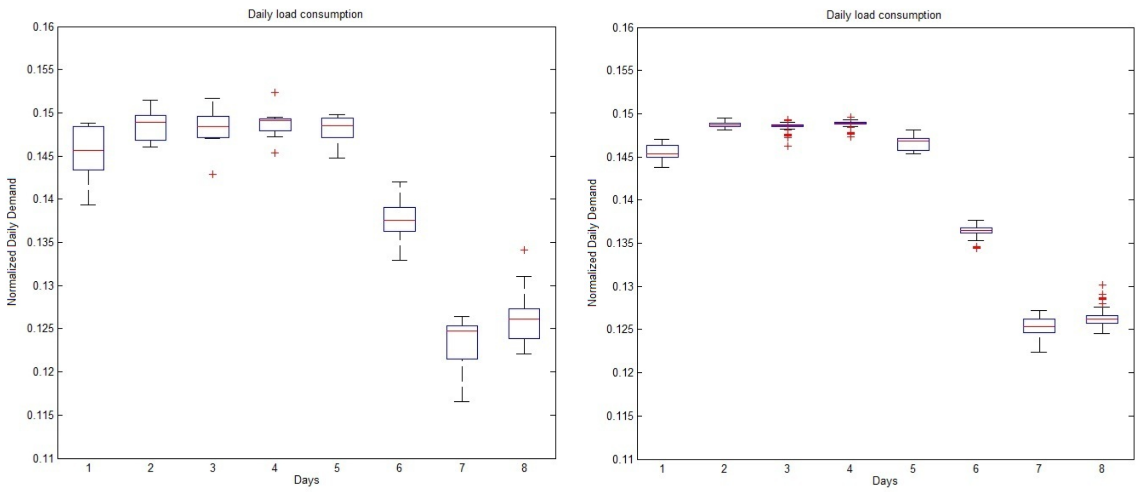
Special Day	**Real Data**	**Virtual Data**
Monday	11	231
Tuesday	19	399
Wednesday	14	294
Thursday	22	462
Friday	10	210
Friday	6	126

Approach 2 ($\lambda = 0.1$)

Special Day	**Real Data**	**Virtual Data**
Monday	11	649
Tuesday	22	1298
Wednesday	14	826
Thursday	22	1298
Friday	19	1121

The two following box plots 4.5 represent the degree of dispersion (spread) and skewness between real and virtual data given in first approach to the Monday holidays. It is possible to verify that ITLMS algorithm converges the virtual data for the mode of the real data.

When dealing with neural networks training it is very important the evaluation of the dispersion and skewness of the training data, once that they will reflect on the final quality of results. Therefore, as the dispersion of the training data on each cluster is low, good results are expected.



(a) Real data of Monday holidays.

(b) Virtual data of Monday holidays.

Figure 4.5: Box plots represent the evolution of the *densification trick* using ITLMS algorithm on group of Monday holidays.

The representation of the densification of all data sets can be seen in appendix A.2.

Chapter 5

Load Forecasting Models

In this chapter all models which were taken into account to perform the load forecasting on special days will be described, as well as the results of each of them.

As mentioned above, all models were based on Autoencoders so in order to perform their training the MATLAB NN Toolbox and their functions was used as will be explained below.

5.1 Training with MATLAB NN Toolbox

The implementation of neural networks has proven to be a complex task, composed of many sub-tasks that must be carefully considered in order to achieve high accuracy. Can be considered sub-tasks like the data normalization (properly dealt in Chapter 4),the choice on the number of hidden layers and neurons, the selection of the activation functions and network training functions, among many others topics and parameters that will be taken into account in this Chapter.

In order to perform the training of the autoencoder (AANN), as outlined above the load forecasting will be based on the daily energy consumption of the days preceding the holiday.

To find the best training model of the autoencoder (AANN), the MATLAB Neural Network Toolbox was used. This Toolbox contains many network training functions, so after various experiences the two training functions which achieved the best results were selected, Levenberg-Marquardt backpropagation (*trainlm*) and bayesian regulation backpropagation (*trainbr*). These two training functions will use the MSE performance function. Its results can be seen further below.

5.1.1 Levenberg-Marquardt backpropagation

trainlm is a network training function that updates weight and bias values according to Levenberg-Marquardt Algorithm (LMA) optimization. It is often the fastest backpropagation algorithm in the MATLAB toolbox, although it does require more memory than other algorithms [5].

The LMA is a method of optimization firstly published by Kenneth Levenberg [121] and then improved by Donald Marquardt [122].

The LMA method is a standard technique used to solve nonlinear least squares problems. This optimization method is indeed a combination of two minimization methods: the gradient descent method and the Gauss-Newton method.

In the gradient descent method, the sum of the squared errors is reduced by updating the parameters in the direction of the greatest reduction of the least squares objective. In the Gauss-Newton method, the sum of the squared errors is reduced by assuming the least squares function is locally quadratic, and finding the minimum of the quadratic.

Therefore, LMA method brings together the best from both methods, acts more like a gradient-descent method when the parameters are far from their optimal value, and acts more like the Gauss-Newton method when the parameters are close to their optimal value [123].

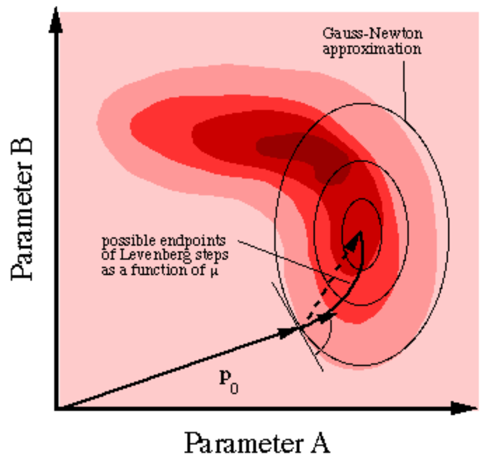


Figure 5.1: Levenberg-Marquardt Algorithm.

The LMA (like Newton's methods) was designed to approach second-order training speed without having to compute the Hessian matrix. When the performance function has the form of a sum of squares (as is typical in training neural networks), then the Hessian matrix can be approximated as

$$H = J^T J \tag{5.1}$$

and the gradient will be

$$g = J^T e \tag{5.2}$$

where J is the Jacobian matrix that contains first derivatives of the network errors with respect to the weights and biases, and e is a vector of network errors. The Jacobian matrix can be computed through a standard backpropagation technique that is much less complex than computing the Hessian matrix [124].

The LMA adaptively varies the parameter updates between the gradient descent and Gauss-Newton update using this approximate Hessian matrix,

$$x_{k+1} = x_k - [J^T J + \mu I]^{-1} J^T e \tag{5.3}$$

where I is the identity matrix and μ is the Marquardt adjustment parameter.

Small values of the algorithmic parameter μ result in a Gauss-Newton update and large values of μ result in a gradient descent update. At a large distance from the function minimum, the steepest descent method is utilized to provide steady and convergent progress toward the solution. As the solution approaches the minimum, μ is adaptively decreased, the Levenberg-Marquardt method approaches the Gauss-Newton method, and the solution typically converges rapidly to the minimum [5, 123].

Validation vectors are used to stop training early if the network performance on the validation vectors fails to improve or remains the same for maximum validation failures epochs in a row. Test vectors are used as a further check that the network is generalizing well, but do not have any effect on training [5].

The NN training stops when any of these conditions occur:

- The maximum number of epochs (repetitions) is reached;
- The maximum amount of time is exceeded;
- Performance is minimized to the goal;
- The performance gradient falls below the minimum stipulated;
- Marquardt adjustment parameter μ exceeds the maximum defined;
- Validation performance has increased more than maximum validation failures.

It is important to note that, this function uses the Jacobian, which assumes that performance is a mean or sum of squared errors. Therefore, networks trained with this function must use either the MSE or SSE performance function.

5.1.2 Bayesian regulation backpropagation

trainbr is a network training function that updates the weight and bias values according to Levenberg-Marquardt optimization. It minimizes a combination of squared errors and weights, and then determines the correct combination so as to produce a network that generalizes well. The process is called Bayesian regularization [5].

With this regularization, any oversized network should be able to sufficiently represent the true function.

Considering the application of MacKay's Bayesian techniques [125, 126] will allow the optimal setting of the regularization parameters and it will be possible to prevent overfitting in neural network training [127].

As occurs in *trainlm*, this function uses the Jacobian for calculations, thus the performance function considered should be MSE or SSE.

This Bayesian regularization takes place within the Levenberg-Marquardt algorithm. Backpropagation is used to calculate the Jacobian jX of performance with respect to the weight and bias variables X. Each variable is adjusted according to LMA,

$$jj = jX * jX \tag{5.4}$$

$$je = jX * E \tag{5.5}$$

$$dX = -(jj + \mu I) \tag{5.6}$$

where E is all errors and I is the identity matrix.

For more details the papers of Mackay [125, 126] and Foresee and Hagan [127] can be consulted.

The NN training stops when any of these conditions occur:

- The maximum number of epochs (repetitions) is reached;
- The maximum amount of time is exceeded;
- Performance is minimized to the goal;
- The performance gradient falls below the minimum stipulated;
- Marquardt adjustment parameter μ exceeds the maximum defined;
- Validation performance has increased more than maximum validation failures.

5.1.3 Division of data set

The data set is usually divided in two or three sets. In this thesis will be considered the division in three parts: train, test and validation. The training set is implemented to built up the model adjusting the weights on NN, while test set is to prevent overfitting and validation set is to validate the model built determining how well the predictive model generalizes.

In all models of autoencoder training the following division of data of train, test and validation was taken into account:

Train and Test Virtual data (was divided considering as train data all virtual data least the same number of real data, in order to applied the last number of virtual data as test data);

Validation Real data.

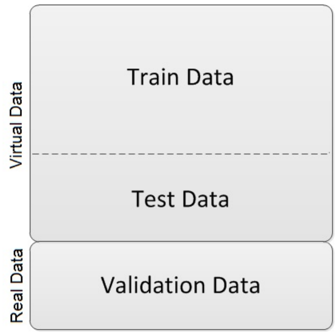


Figure 5.2: Division of data set.

5.1.4 Autoencoders structures and others relevant parameters

In this study, three different autoencoders structures will be considered:

1. The first autoencoder structure is composed by eight-input, eight-output and seven neurons in the hidden layer;
2. The second is composed by seven-input, seven-output and six neurons in the hidden layer;
3. The third is composed by five-input, five-output and four neurons in the hidden layer.

It is important to note that the last neurons in each input and output configuration correspond to the special day, the other inputs / outputs correspond to the days preceding the holiday. Therefore, these three structures are considered in way to verify the influence of more or less days in the forecasting model. It will be analysed only these three structures, once to evaluate all possible structures would require an exhaustive study, this would require more time than we had.

The following picture illustrate the general autoencoder structure.

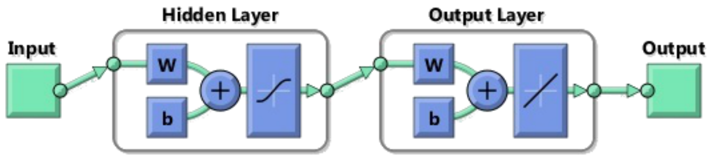


Figure 5.3: Autoencoder structure, from MATLAB.

As shown by the picture, and was presented in subchapter 3.2, the autoencoder needs two transfer functions for "encoding" and "decoding", i.e., a compressing operation transforming from data space to code space at the hidden layer, and reverse transformation from code space to data space at the output layer.

Transfer functions or activation functions, φ at 3.22, are used for limiting the amplitude of the output of a neuron.

The "encoding" transfer function was performed by a hyperbolic tangent sigmoid (*tansig*) and is given by

$$a \;=\; tansig(n) \;=\; 2/\left(1+e^{(-2\,\cdot\,n)}\right)-1 \tag{5.7}$$

where n represents input data and a output data.

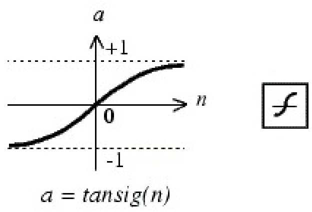


Figure 5.4: Hyperbolic tangent sigmoid transfer function, from [5].

The "decoding" was performed by a linear transfer function (*purelin*) and is given by

$$a \;=\; purelin(n) \;=\; n \tag{5.8}$$

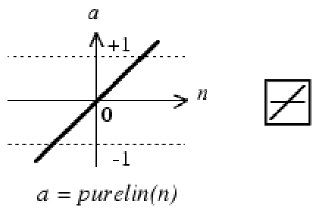


Figure 5.5: Linear transfer function, from [5].

The following figure illustrate an example of the window of the results of training the autoencoder with the MATLAB NN Toolbox. In this example the cluster of Thursday holidays of the first approach of the *densification trick* using ITLMS algorithm and the function Levenberg-Marquardt backpropagation (*trainlm*) as the network training function were used.

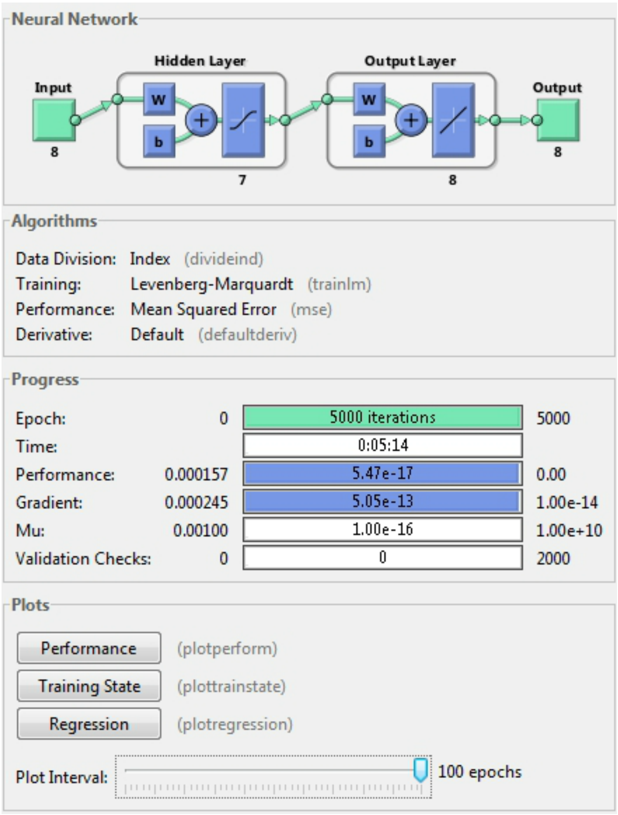


Figure 5.6: Neural network training window, from MATLAB.

In this figure it is possible to observe the principal parameters which were taken into account to make the autoencoder training:

Table 5.1: Network training functions parameters.

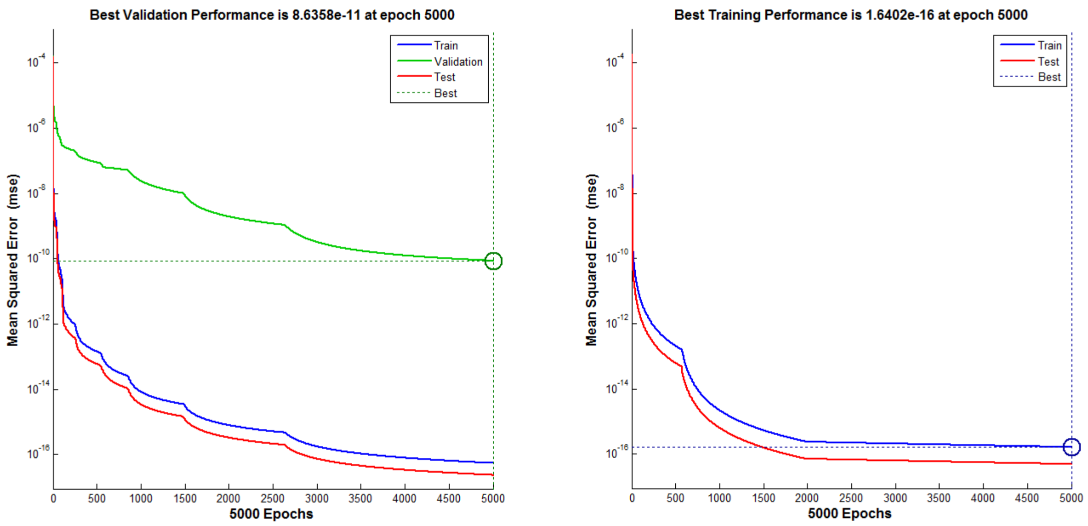
Maximum number of epochs	5000
Performance goal	0
Maximum validation failures	2000
Minimum performance gradient	1×10^{-14}

Note: These parameters were chosen after many tests and simulations, is possible achieve better results with different parameters, but this would require an exhaustive study.

Besides of these parameters, to perform the diverse training models the selection of the training function (*trainlm* or *trainbr*) and the structure of the autoencoder (8-7-8, 7-6-7 or 5-4-5) were also taken into account. In the example of the figure 5.6 can be seen how the choice of all these parameters was made.

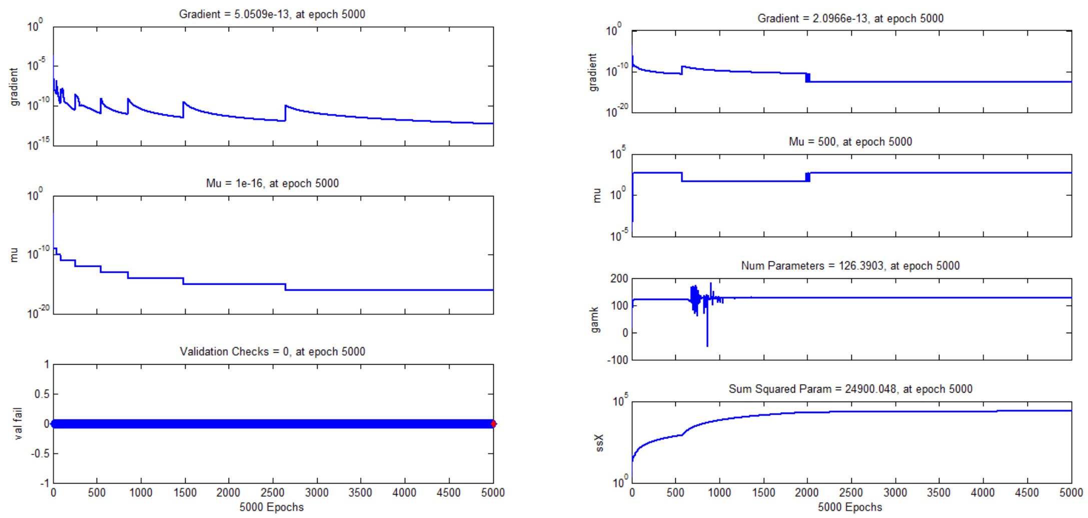
Another important observation in this example is the NN training which was stopped when the maximum number of epochs were reached.

The MATLAB NN Toolbox gives also more results to evaluate the performance of training models. Then the comparative results of the two training functions (trainlm and trainbr) with the autoencoder structure 8-7-8 under the same conditions will be presented.



(a) *trainlm (training performance MSE=5.47×10*$^{-17}$*)* (b) *trainbr (training performance MSE=1.64×10*$^{-16}$*)*

Figure 5.7: Performance plots, from MATLAB.



(a) *trainlm* (b) *trainbr*

Figure 5.8: Training state plots, from MATLAB.

Analysing the results is always important evaluate if an overfitting problem occurs at the process of training the NN.

A model is typically trained by maximizing its performance on some set of training data. By the other hand, the model efficacy is determined not by its performance on the training data but by its ability to perform well on unseen data.

When a model begins to memorize training data rather than learning to generalize from trend this is called overfitting. This corresponds a test data error much higher than the train data error and means that the neural system is over determined [6]. A properly trained system should correspond with the same order of magnitude for the error measures to both training and testing data. To avoid overfitting, this point must be identified and the training must be stoped. The following figure ilustrates the point where the optimal learning and generalization are achieved, that is close to the global minimum of test error.

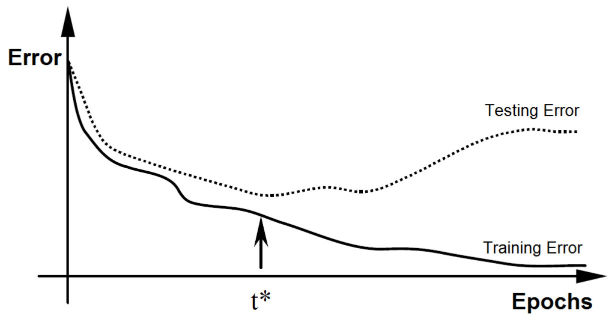


Figure 5.9: Overfitting, overfitting, from the training epoch t^*, from [6].

Therefore, through the results of the figure 5.7 is possible verify that the problem of overfitting does not occur.

Despite the results indicate that the *trainlm* is the training function with minor training performance MSE, the evaluation of the best method of training autoencoders is only possible to verify further along once that only the results of the complete model can demonstrate the correct validation of the predictions.

5.2 Description of the Forecasting Models

In this section the complete models which were used to perform the forecasts will be described. Taking into account the base models presented in the section 3.3 the following three models will be considered:

Model 1 – Unconstrained search model - an optimization algorithm searches for the input values that minimize the input/output error on the signal that correspond to the special day;

Model 2 – Unconstrained search model - an optimization algorithm searches for the input values that minimize the input/output error on all the signals except one that correspond to the special day;

Model 3 – Constrained search model - an optimization algorithm searches for the input values that minimize the input/output error on all the signals.

The following pictures illustrate well these three forecasting models.

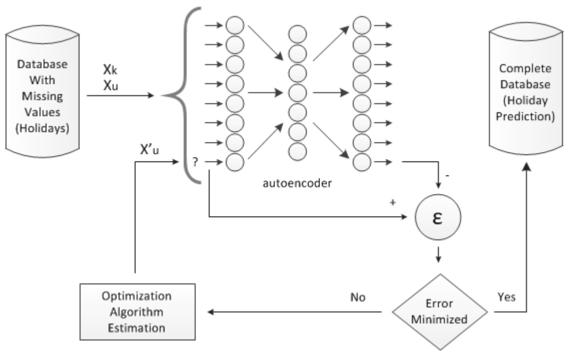


Figure 5.10: Model 1 – Autoencoder and optimization algorithm - Unconstrained Search model.

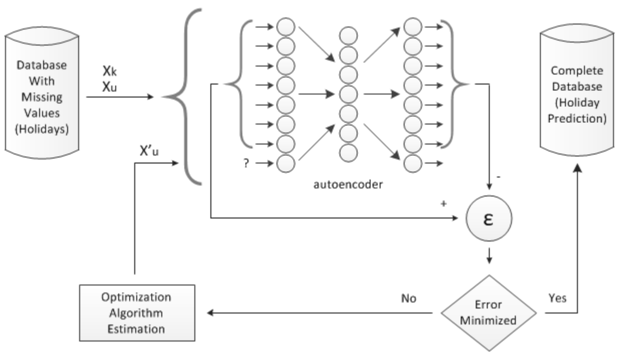


Figure 5.11: Model 2 – Autoencoder and optimization algorithm - Unconstrained Search model.

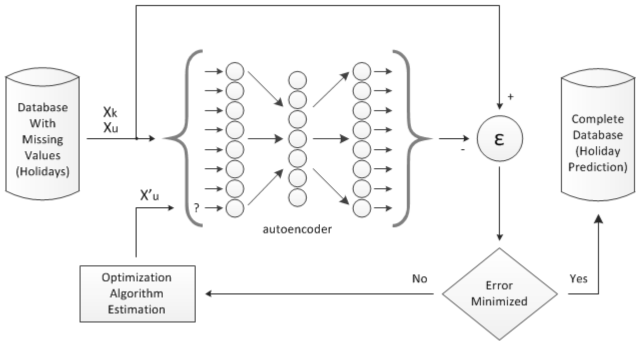
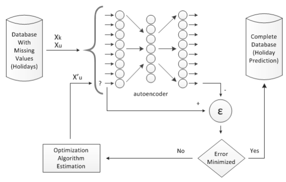
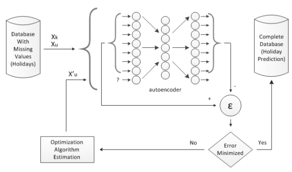
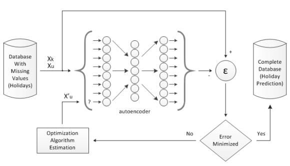


Figure 5.12: Model 3 – Autoencoder and optimization algorithm - Constrained Search model.

Therefore, it is concluded that, complete forecasting models can be divided as follows.

Table 5.2: Forecasting models, complete description.

Model	Optimization search model	Days preceding the holiday	Training functions
1		4	trainlm
2			trainbr
3		6	trainlm
4			trainbr
5		7	trainlm
6			trainbr
7		4	trainlm
8			trainbr
9		6	trainlm
10			trainbr
11		7	trainlm
12			trainbr
13		4	trainlm
14			trainbr
15		6	trainlm
16			trainbr
17		7	trainlm
18			trainbr

5.3 Results analysis

In this section the results of the models under analysis will be presented, as well as the relating discussion of them.

Before being assessed all forecasting models, the metaheuristics PSO and EPSO were evaluated in order to find which is the most robust.

5.3.1 PSO vs EPSO

In order to evaluate the most robust metaheuristic method to implement in forecasting model, some tests were performed.

In these tests only one forecasting model was considered and the results given by PSO and EPSO in the same conditions.

The forecasting model was

- Training function – trainbr;
- Horizon of days preceding the holiday – 7 days;
- Optimization search model – model 3.

The initialization parameters of the two metaheristics are

Table 5.3: Parameters initialization.

PSO

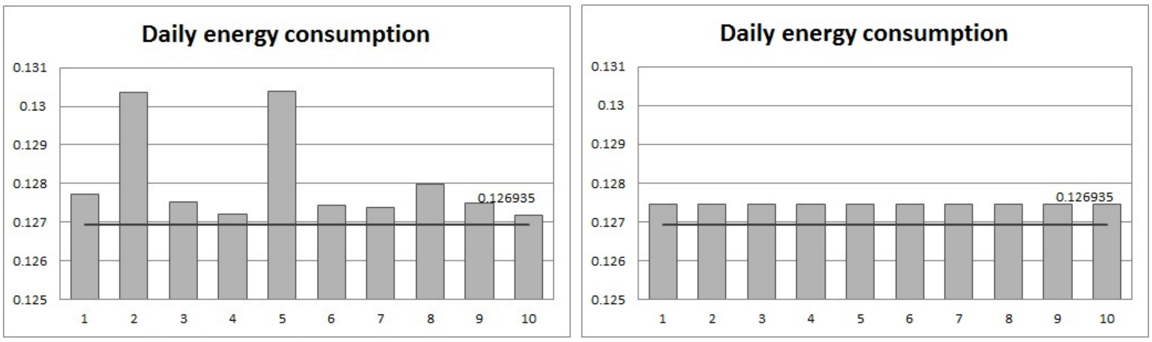
Parameters	**Value**
W_m	0.03
W_c	0.06
Iterations	200

EPSO

Parameters	**Value**
W_i	0.03
W_m	0.03
W_c	0.06
W_b	0.06
τ	0.01
P	0.8
Iterations	200

Note: These parameters initialization were chosen after many tests and simulations, is possible achieve better results with different parameters, but this would require an exhaustive study.

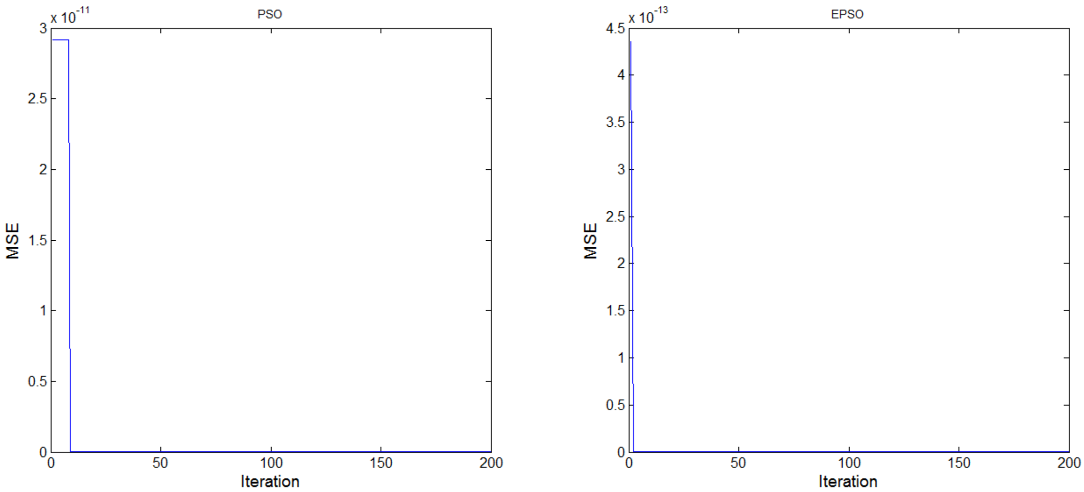
Therefore, the results after ten forecasts of the first real data of Thursday holidays cluster (normalized daily energy consumption = 0.126935) are illustrated in the next figures.



(a) PSO (b) EPSO

Figure 5.13: Ten forecasts of the same day using PSO and EPSO as optimization algorithm.

The following pictures illustrate an example of the MSE minimization which were performed by PSO and EPSO algorithms.



(a) PSO (b) EPSO

Figure 5.14: MSE minimization using PSO and EPSO as optimization algorithm.

Given these results, it can be concluded that EPSO proved to be a metaheuristic more robust than the PSO.

The algorithm robustness, which has to do with the warranty (probability) that, regardless of the initialization, the algorithm will converge to the optimal or your vicinity. It is not expected that the algorithm is executed several times on the same problem. It is expected that it gives only one trust result. It is precisely this confidence in the result which is measured by the concept of robustness, it is expected that the algorithm when, executed several times, always finds good results with very small deviations of the optimal solution.

Analyzing the results in detail, it was found that the PSO, in many cases, gets stuck in local optima and that this influences the quality of results.

Furthermore, EPSO has a better accuracy for the same computational effort.

Therefore, the optimization method chosen is the EPSO.

5.4 Prediction results

The tests of the all models were performed in four stages:

1. With the first approach of the densification of data and the application of only one cluster (e.g. cluster of Thursday holidays) all 18 forecasting models were tested. The intention with this study is verify the influence of more or less days in the forecasting model;

2. After the evaluation the results of the first stage the best models (consideration of more or less days preceding the holiday) were considered to perform the forecasting on all clusters of the first approach of data densification;

3. For the best model found in the stage 2, the influence of the consideration of two different approachs of densification of data was evaluated (Approach 1: 21 virtual data for each real data; Approach 2: 59 virtual data for each real data).

4. In the end, the comparison with prediction results given in [4] will be made.

The results analisys will be evaluated by some forecasting indicators such as the variation range std (standard deviation), MSE (mean square error), MAE (mean absolute error), NMSE (normalized mean square error), NMAE (normalized mean absolute error) and MAPE (mean absolute percentage error). These performance metrics and their calculations are shown in the following table.

Table 5.4: Performance metrics and their calculations.

Metrics	Calculation
std	$std\left(x_i\right) = \frac{\sum_{i=1}^{n}\left(x_i - \overline{x}\right)}{n-1}$
MSE	$MSE = \frac{\sum_{i=1}^{n}\left(x_i - \hat{x}_i\right)^2}{n}$
MAE	$MAE = \frac{\sum_{i=1}^{n}\left\lvert x_i - \hat{x}_i\right\rvert}{n}$
NMSE	$NMSE = \frac{\sum_{i=1}^{n}\left(x_i - \hat{x}_i\right)^2}{std(\hat{x}_i)\cdot n}$
NMAE	$NMAE = \frac{\sum_{i=1}^{n}\left\lvert x_i - \hat{x}_i\right\rvert}{\sum_{i=1}^{n}\left\lvert x_i\right\rvert}$
MAPE	$MAPE = \frac{\sum_{i=1}^{n}\left\lvert x_i - \hat{x}_i\right\rvert / x_i}{n} \times 100\%$

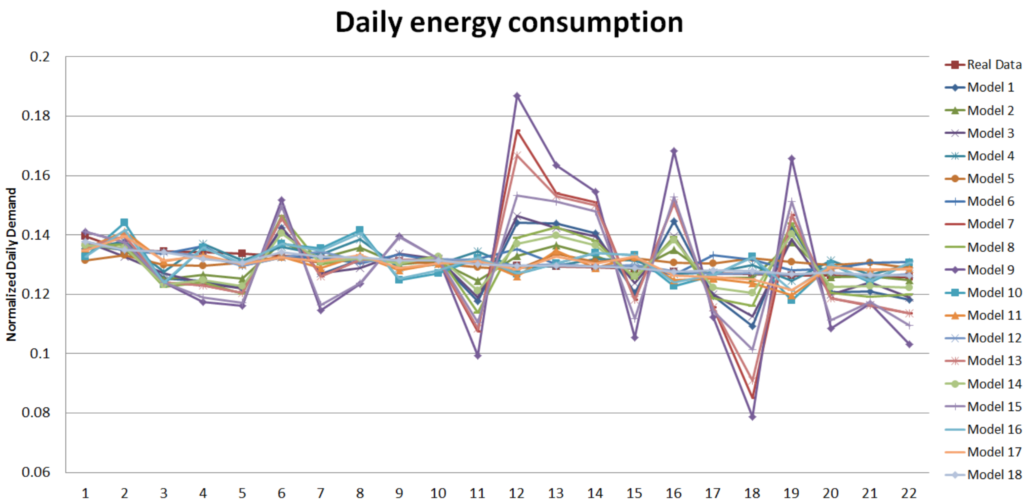
* x_i and $\hat{x}_i$ are the real values and predicted values.

5.4.1 Stage 1

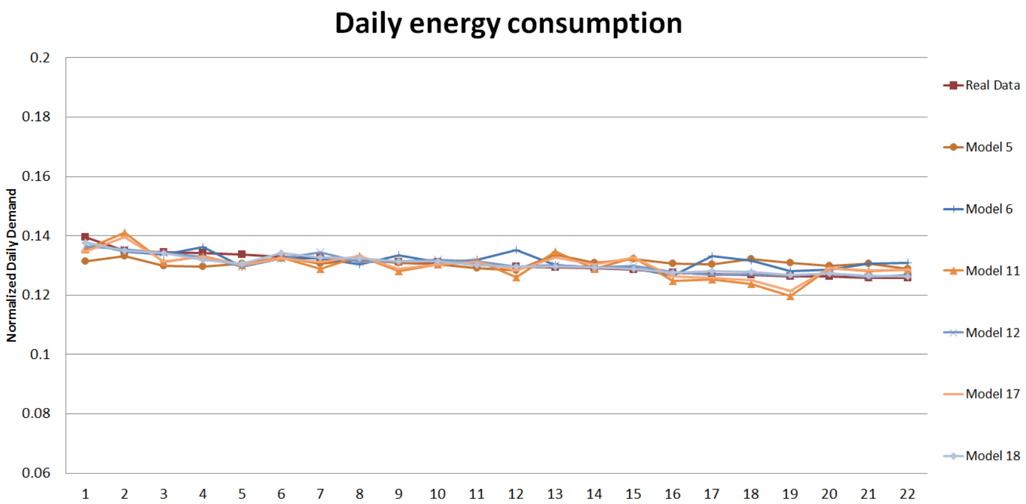
In this first stage, it is intended to make the evaluation of all forecasting models. After all tests the following results were reached.

Table 5.5: Prediction results summary of all forecasting models (Thursday).

Model	std	MSE	MAE	NMSE	NMAE	MAPE
1	1.058%	1.0E-04	0.871%	2.865%	6.675%	6.719%
2	0.483%	2.5E-05	0.372%	0.699%	2.850%	2.841%
3	0.944%	8.2E-05	0.783%	2.261%	6.006%	6.036%
4	0.384%	1.0E-05	0.267%	0.288%	2.047%	2.037%
5	0.138%	1.3E-05	0.298%	0.350%	2.282%	2.282%
6	0.277%	8.6E-06	0.221%	0.238%	1.712%	1.712%
7	1.929%	3.4E-04	1.433%	9.512%	10.991%	11.093%
8	0.984%	9.0E-05	0.800%	2.487%	6.134%	6.161%
9	2.669%	6.7E-04	2.122%	18.532%	16.272%	16.438%
10	0.625%	3.0E-05	0.470%	0.823%	3.607%	3.592%
11	0.453%	1.1E-05	0.288%	0.308%	2.208%	2.209%
12	0.306%	1.7E-06	0.090%	0.048%	0.691%	0.681%
13	1.757%	2.9E-04	1.352%	7.880%	10.368%	10.462%
14	0.724%	5.1E-05	0.590%	1.404%	4.523%	4.535%
15	1.708%	2.7E-04	1.476%	7.490%	11.317%	11.406%
16	0.537%	2.3E-05	0.415%	0.636%	3.183%	3.168%
17	0.373%	7.3E-06	0.233%	0.202%	1.788%	1.786%
18	0.304%	1.1E-06	0.071%	0.031%	0.544%	0.540%



(a) All models.



(b) Six best models.

Figure 5.15: Prediction results to all 22 Thursday holidays.

Note: The special days in this graph are not distributed by a temporal logic but rather by the value of daily energy, from larger to smaller.

It is possible verified that have many models with predictions quite far the true values. The best models at the issue of number of days that preceding the special day, are those which consider 7 days preceding the holiday.

This occurs because the consideration of more days allows a better differentiation of the patterns and their weekly behavior.

Therefore, these six models will be analysed in more detail in the next stage.

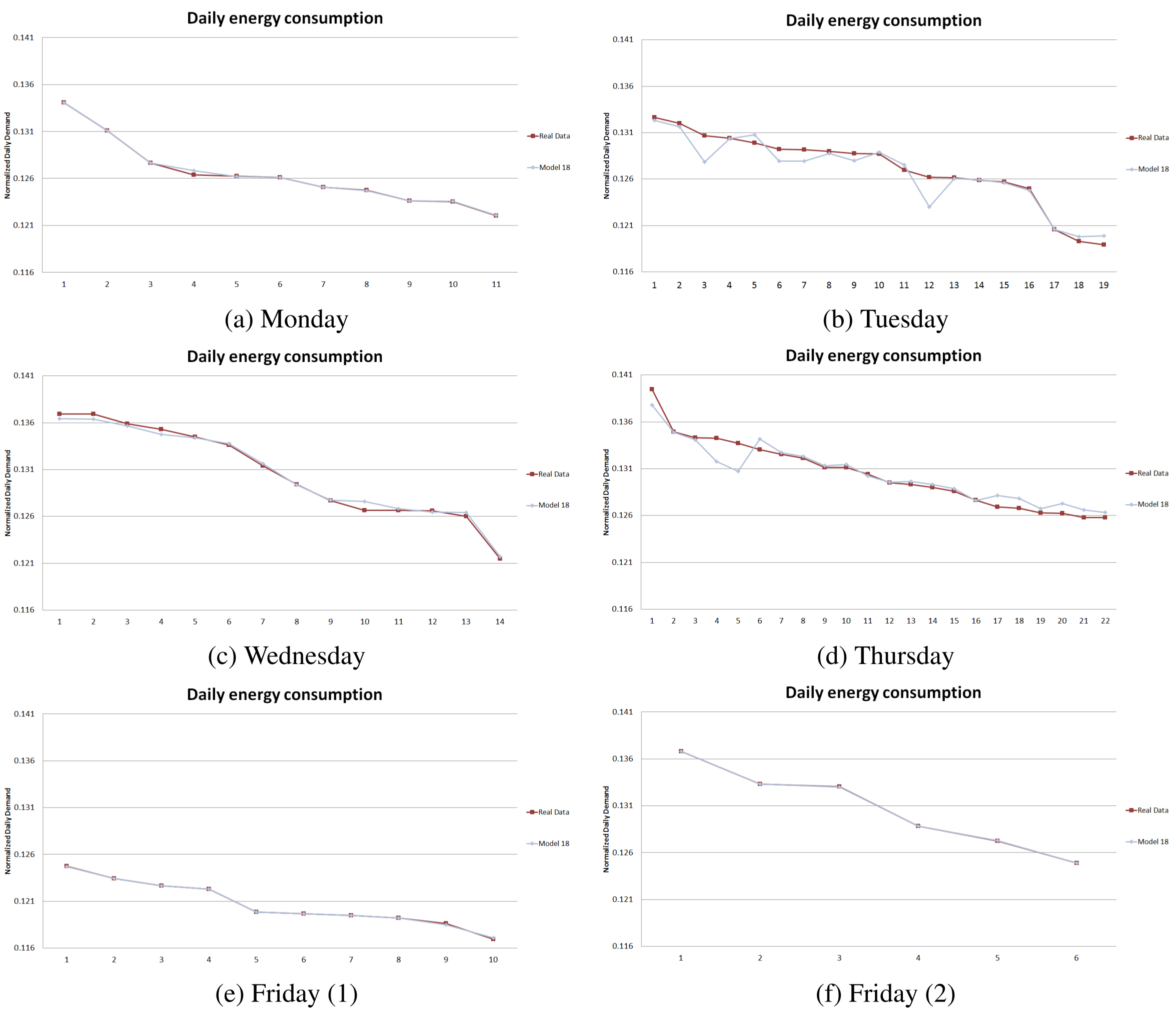
5.4.2 Stage 2

Once found the best models, their results to all clusters of special days will be analysed.

All these results can be seen in appendix B.0.3. In this analysis only will be considered in more detail the best model of all, the Model 18. The results can be seen below:

Table 5.6: Prediction results summary of the Model 18.

Special day	Real days tested	std	MSE	MAE	NMSE	NMAE	MAPE
Monday	11	0.349%	2.0E-08	0.007%	0.001%	0.053%	0.054%
Tuesday	19	0.375%	1.3E-06	0.073%	0.033%	0.571%	0.569%
Wednesday	14	0.466%	1.6E-07	0.031%	0.003%	0.234%	0.234%
Thursday	22	0.304%	1.1E-06	0.071%	0.031%	0.544%	0.540%
Friday (1)	10	0.243%	3.4E-09	0.003%	0.0001%	0.027%	0.028%
Friday (2)	6	0.444%	2.0E-10	0.001%	4.4E-06%	0.007%	0.007%



(a) Monday (b) Tuesday (c) Wednesday (d) Thursday (e) Friday (1) (f) Friday (2)

Figure 5.16: Prediction results summary of the Model 18 applied to all clusters.

With this model excellent results were achieved. Through the graphs is possible to observe that the prediction results were close enough of the real results.

The Normalized Mean Absolute Error (NMAE) varies from 0.007% (Friday (2)) to 0.571% (Tuesday), while the variation range of the corresponding standard deviation is from 0.243% to 0.466%. The indicator Mean Absolute Percentage Error (MAPE) varies from 0.007% to 0.569%. This accuracy is very satisfactory.

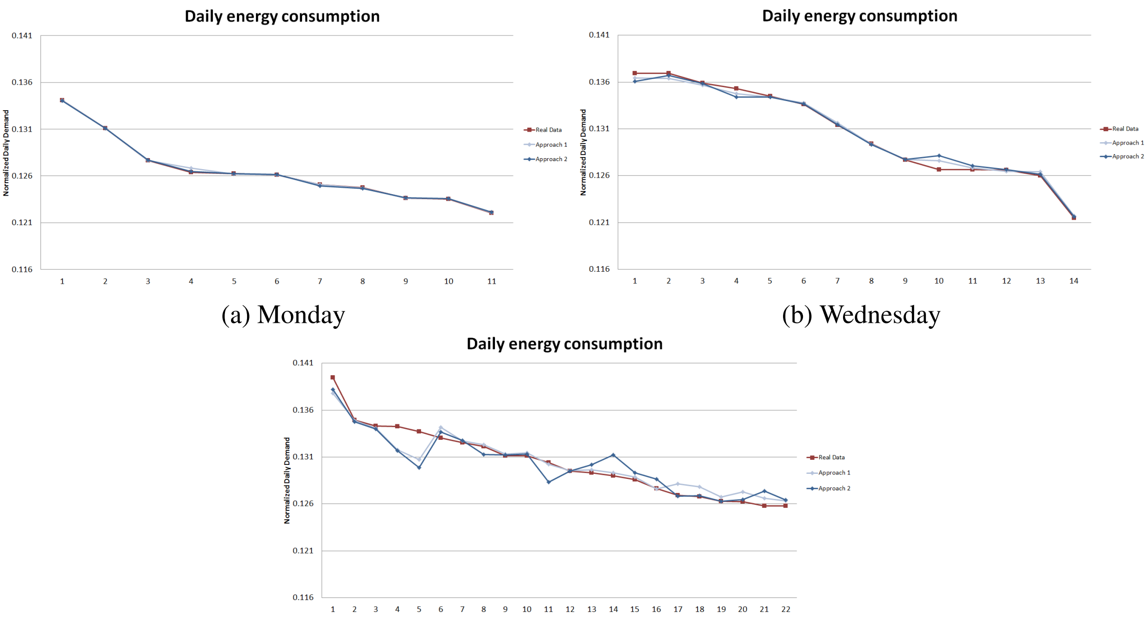
5.4.3 Stage 3

This stage, have the objective of to evaluate the performance of the best model found considering two different approaches of densification of data. Using ITLMS, as stated in chapter 4, two different virtual data sets were achieved (Approach 1: 21 virtual data for each real data; Approach 2: 59 virtual data for each real data).

As the clusters of real data are not equal for the two approaches, this comparison only was made to the three clusters with the same real data. The results achieved were:

Table 5.7: Prediction results summary. Comparison between two different approachs of densification of data (A1 and A2).

Special day	Real days tested	std A1	std A2	NMAE A1	NMAE A2
Monday	11	0.349%	0.349%	0.053%	0.053%
Wednesday	14	0.466%	0.463%	0.234%	0.254%
Thursday	22	0.312%	0.312%	0.544%	0.686%



(a) Monday (b) Wednesday

(c) Thursday

Figure 5.17: Prediction results of the two approaches.

It is possible conclude that this two approaches are equivalents. For the Mondays predictions the same forecasting indicators were obtained. In the other two cluster the results were also very close.

5.4.4 Stage 4

In this final stage, the results achieved in the paper [4] will be compared with the results that were given by the best model found in this work, considering, logically, the same clusters of data.

In [4] were used the same data sets of the second approach of the densification of data performed in this dissertation.

In order to provide a fuller understanding of the differences between this two forecasting methods, the following table presents the characteristics of both techniques.

Table 5.8: Characteristics of two different forecasting methods. Proposed in this work and proposed in the paper [4].

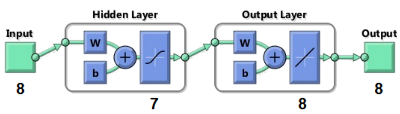
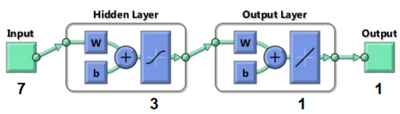
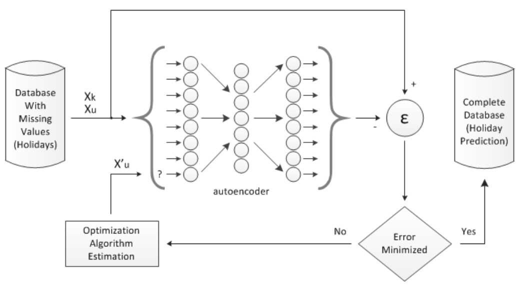
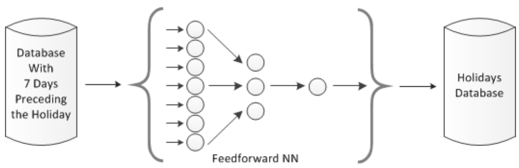
Based on Autoencoders	**Based on Simple Feedforward NN [4]**
Network Training Structure	
 8-7-8	 7-3-1
Transfer functions	
tansig – purelin	*tansig – purelin*
Training functions	
Bayesian regulation backpropagation	Simple Backpropagation
Forecasting Model	
	

Table 5.9: Prediction results summary. Comparison with the results achieved in [4]

Special day	Real days tested	std	std in [4]	NMAE	NMAE in [4]
Monday	11	0.35%	1.76%	0.05%	1.85%
Tuesday	22	0.45%	2.50%	0.91%	2.82%
Wednesday	14	0.46%	1.90%	0.25%	3.41%
Thursday	22	0.31%	1.66%	0.69%	2.55%
Friday	19	0.50%	2.46%	1.08%	3.92%

The Normalized Mean Absolute Error (NMAE) varies from 0.05% (Monday) to 1.08% (Friday) for the model with Autoencoder, while the results in [4] were in the order of 1.85% (Monday) to 3.92% (Friday).

The variation range of the corresponding standard deviation is from 0.31% to 0.50% better than in [4] with 1.66% to 2.50%.

As can be demonstrated, with this new forecasting model into analysis, better results were achieved.

An improvement of std prediction parameter in the order of 76% (Wednesday) to 82% (Tuesday) was achieved, and for NMAE in the order of 68% (Tuesday) to 97% (Monday).

To all results was verified that the best results were achieved to the clusters with less days in real data. It is simple verify that this occur because with less data is more simple to find the correct pattern while for a cluster with more patterns becomes more difficult to interpolate the correct pattern.

For more detail of the results achieved the following excel file can be consulted:
https://dl.dropboxusercontent.com/u/23872037/Results.xlsx

Chapter 6

Conclusion

This work demonstrates the advantage of using the Information Theoretic Learning Mean Shift algorithm, in the form of a *desnsification trick*, in order to solve the problem of scarce data on special days.

With this tool became possible the neuronal networks training even when faced with scarce data sets, the problem of special days. The ITLMS can be used to identify distinct clusters in the load data, associating in a easy way holidays that occurs in distinct days of the week and then allow the virtual data collected representing these specific clusters.

It has been proven that the use of the virtual data as training set can be applied as if they were training with real data.

The alliance of this powerful tool with an Auto Associative Neural Network, demonstrated to be a robust model in the load forecasting on special days.

This Autoencoder based on missing data estimation, use an optimization performed by the metaheuristic EPSO in order to predict the special day (missing data) taking in consideration the days that precede the holiday.

The high accuracy achieved by this method confirms that this tools can bring improvements in the performance of the load forecasting methodologies, specially, on days with occurence of scarce historical data to represent their behavior.

The achieved results, considering the NMAE indicator, were in the order of 0.007% (Friday (2)) to 0.571% (Tuesday) in the first approach of data densification (21 virtual data for each real data), and in the order of 0.05% (Monday) to 1.08% (Friday) in the second approach (59 virtual data for each real data).

Through the obtained results, it can be concluded that Autoencoders based on missing data estimation gives better results than a simple Feedforward Neural Network.

An improvement in the predictions in the order of 68% to 97% was achieved.

This topology could be an important step in load forecasting on special days and even on normal days.

6.1 Future Work

In spite of the advances done in this work, much work remains to be done in the area of load forecasting on special days.

It would be important verify the performance of this new forecasting model, with historical data from other power distribution utilities.

It would be also interesting perform the load forecasting not just on holidays but also on normal days.

Besides the several tests that were made, more tests with other parameters in each tool of the model can allow the achievement of better results.

The development a more efficient algorithm to the autoencoder training can bring better results. There are several ones using evolutionary algorithms in the literature, for example in [128] in order to perform the wind power forecasting, the NN training with metaheuristic EPSO was implemented. In the same work as in others [129, 130], were considered others optimization criteria adopting entropy concepts to train the NN, based in mutual information principle [131]. Renyi's Entropy is combined with a Parzen Windows estimation of the error pdf to form the basis of three criteria (MEE, MCC and MEEF) under which neural networks are trained. In some researchs, the results was favourably compared with the traditional MSE criterion.

Appendix A

Results of the data treatment

A.1 Results of the load correction method

A.1.1 Approach 1

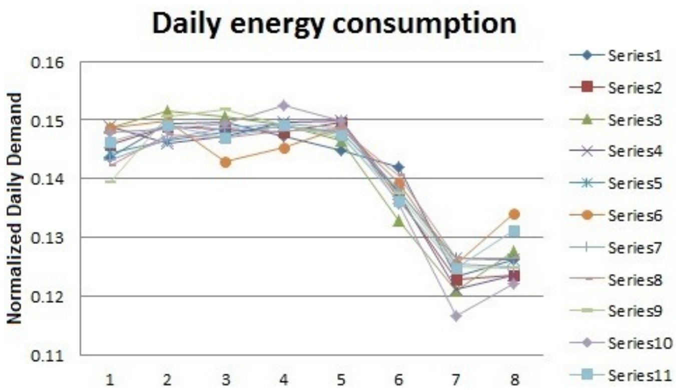


Figure A.1: Monday holidays, cluster with 11 patterns.

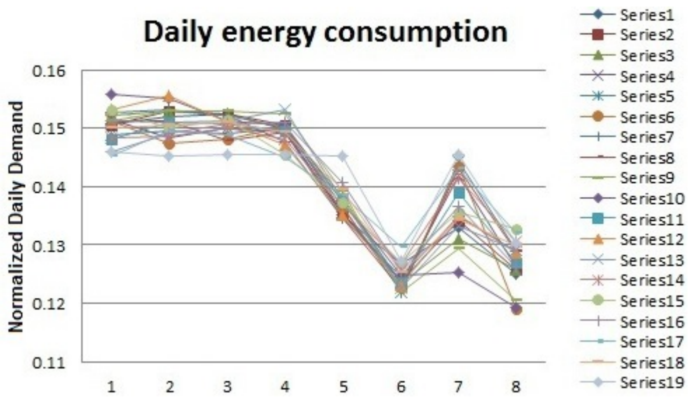


Figure A.2: Tuesday holidays, cluster with 19 patterns.

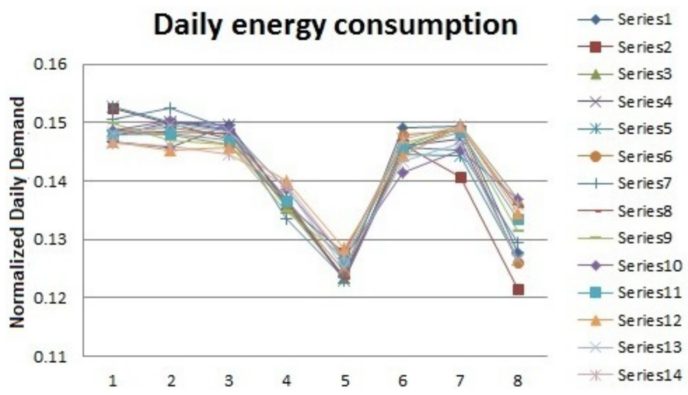


Figure A.3: Wednesday holidays, cluster with 14 patterns.

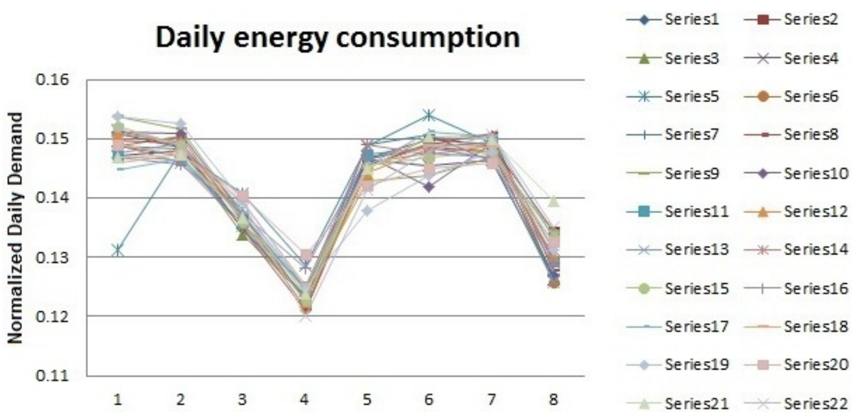


Figure A.4: Thursday holidays, cluster with 22 patterns.

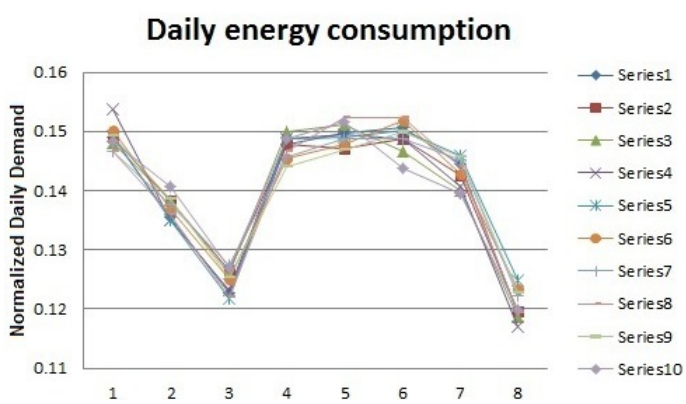


Figure A.5: Friday holidays (1), cluster with 10 patterns.

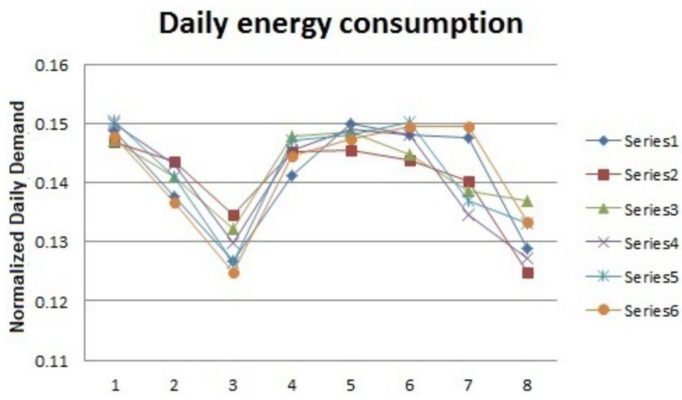


Figure A.6: Friday holidays (2), cluster with 6 patterns.

A.1.2 Approach 2

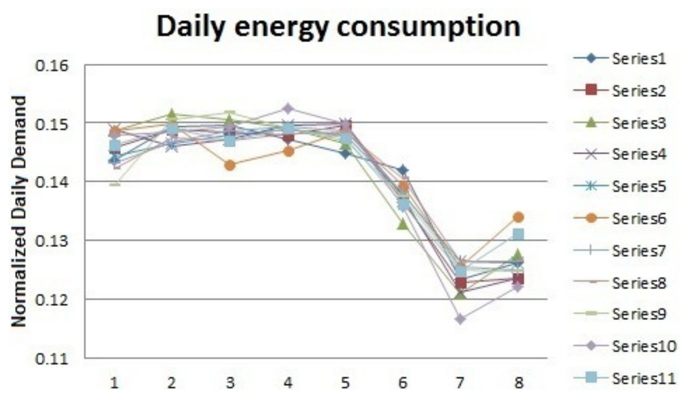


Figure A.7: Monday holidays, cluster with 11 patterns.

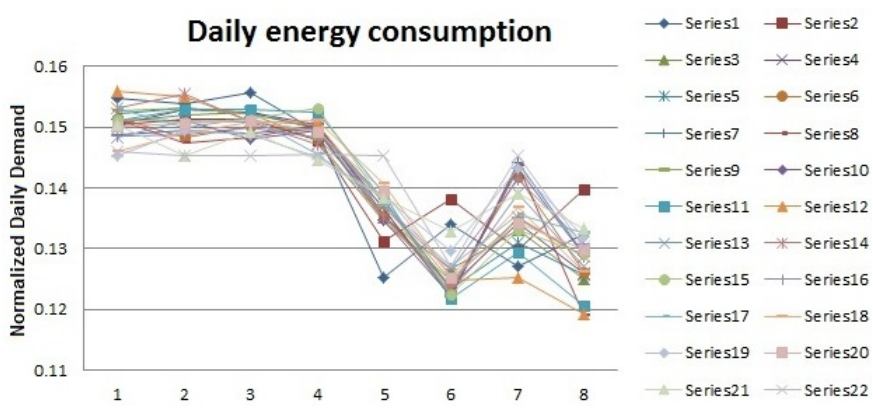


Figure A.8: Tuesday holidays, cluster with 22 patterns.

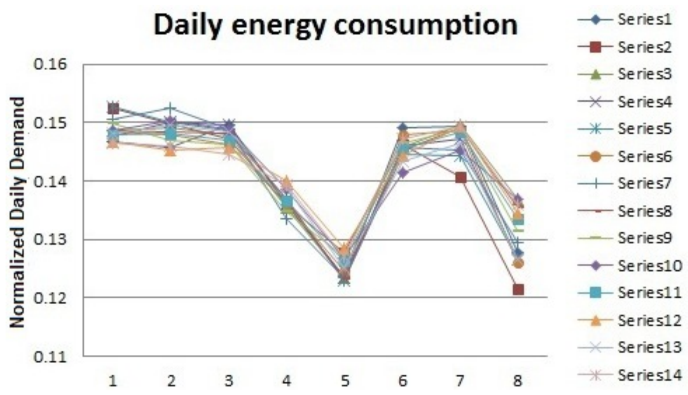


Figure A.9: Wednesday holidays, cluster with 14 patterns.

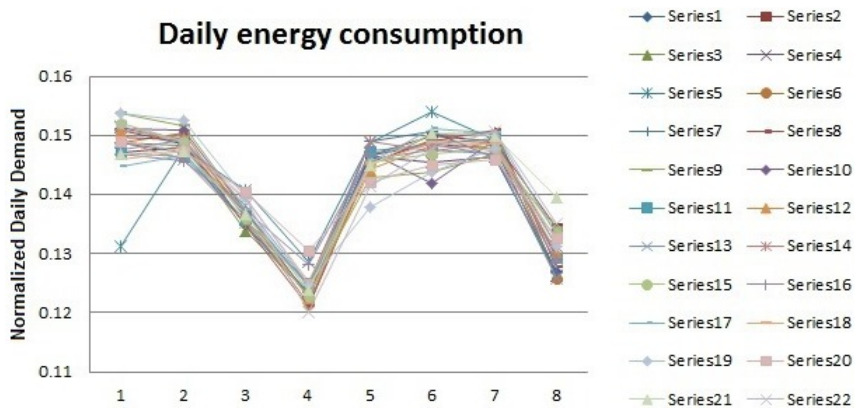


Figure A.10: Thursday holidays, cluster with 22 patterns.

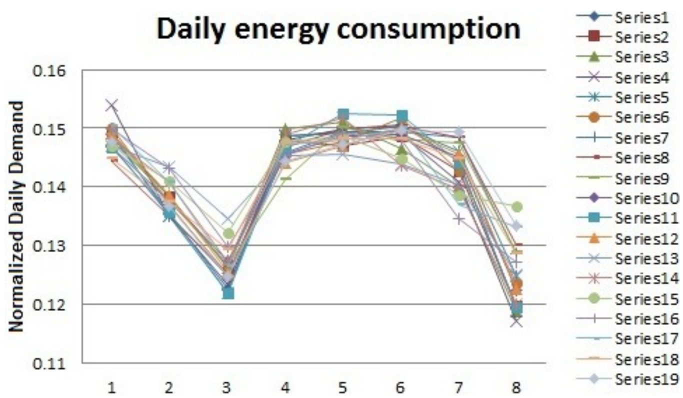
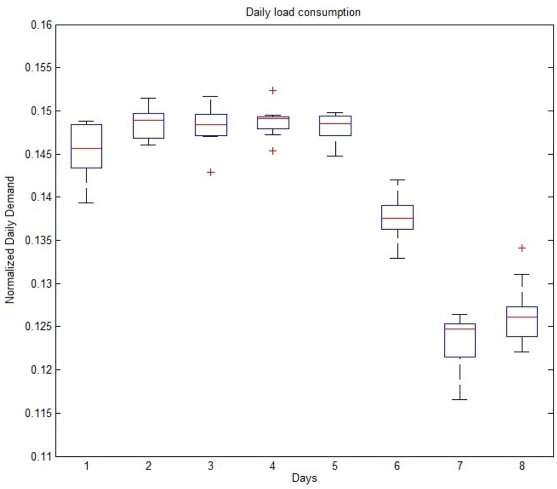


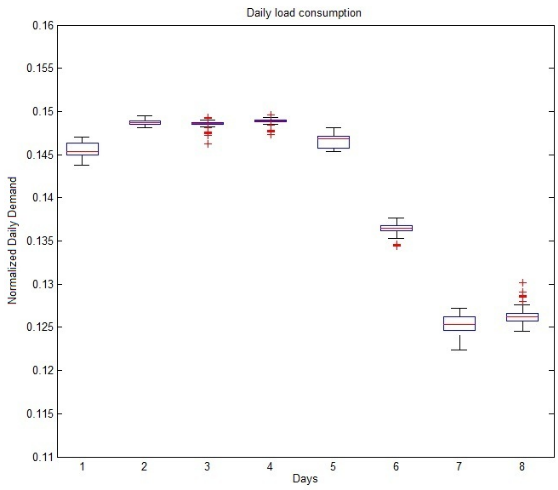
Figure A.11: Friday holidays, cluster with 19 patterns.

A.2 Results of the densification of data sets

A.2.1 Approach 1

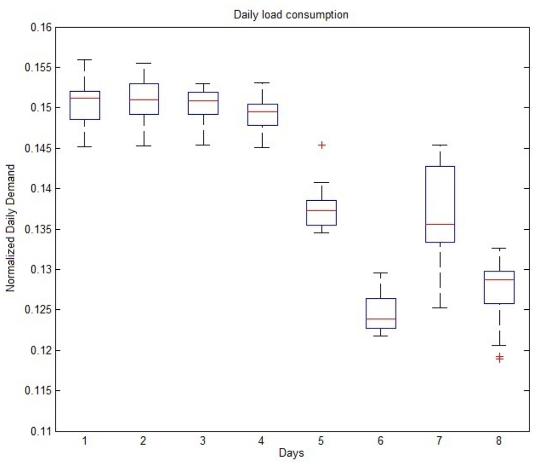


(a) Real data of Monday holidays.

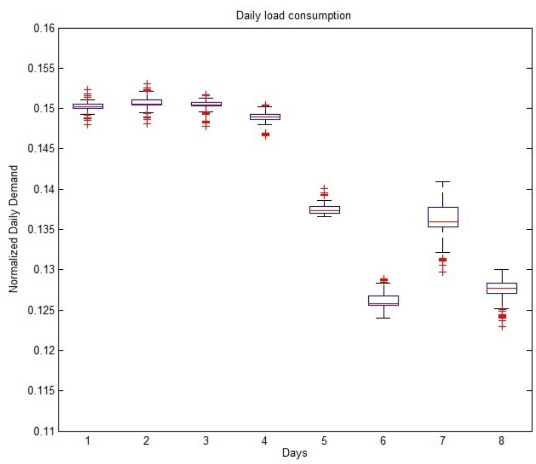


(b) Virtual data of Monday holidays.

Figure A.12: Box plots represent the evolution of the *densification trick* using ITLMS algorithm on group of Monday holidays.

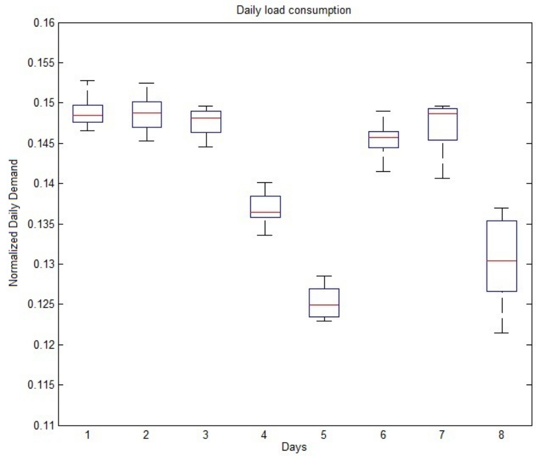


(a) Real data of Tuesday holidays.

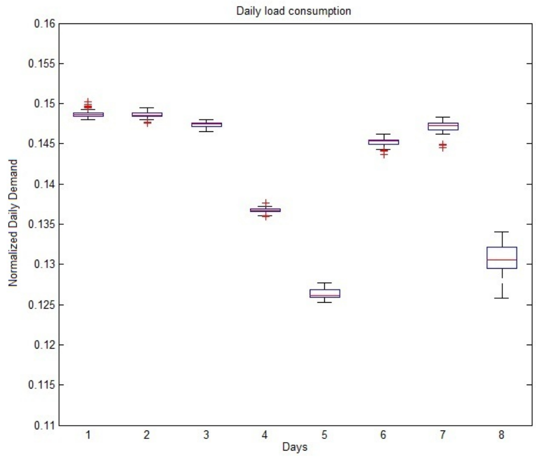


(b) Virtual data of Tuesday holidays.

Figure A.13: Box plots represent the evolution of the *densification trick* using ITLMS algorithm on group of Tuesday holidays.

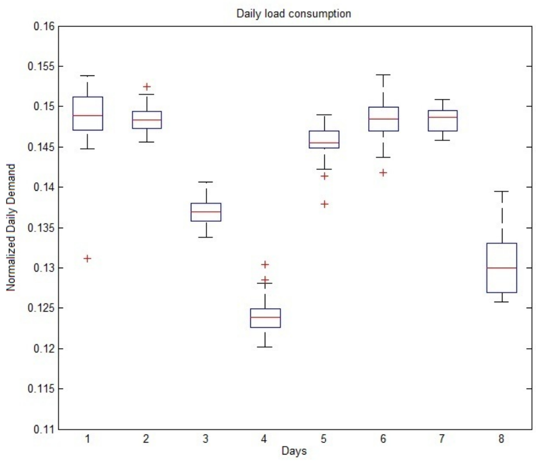


(a) Real data of Wednesday holidays.

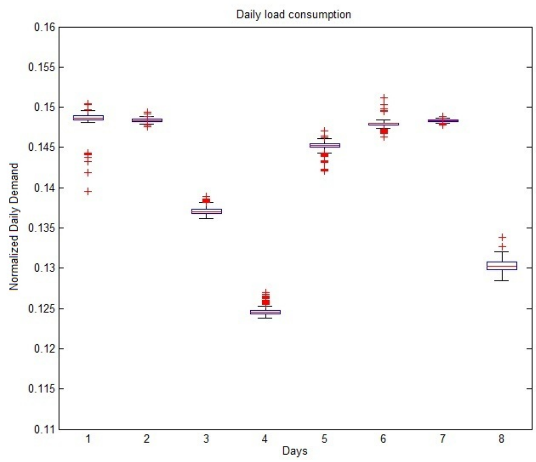


(b) Virtual data of Wednesday holidays.

Figure A.14: Box plots represent the evolution of the *densification trick* using ITLMS algorithm on group of Wednesday holidays.

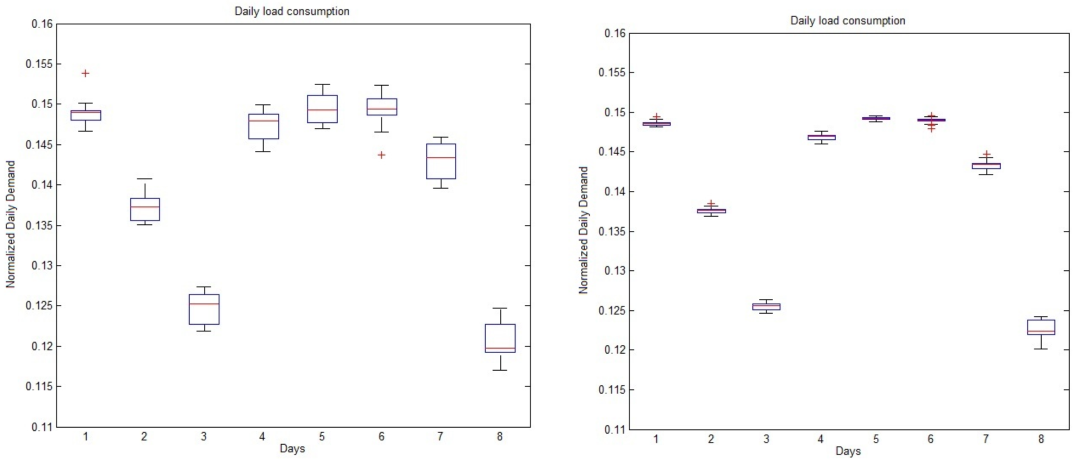


(a) Real data of Thursday holidays.



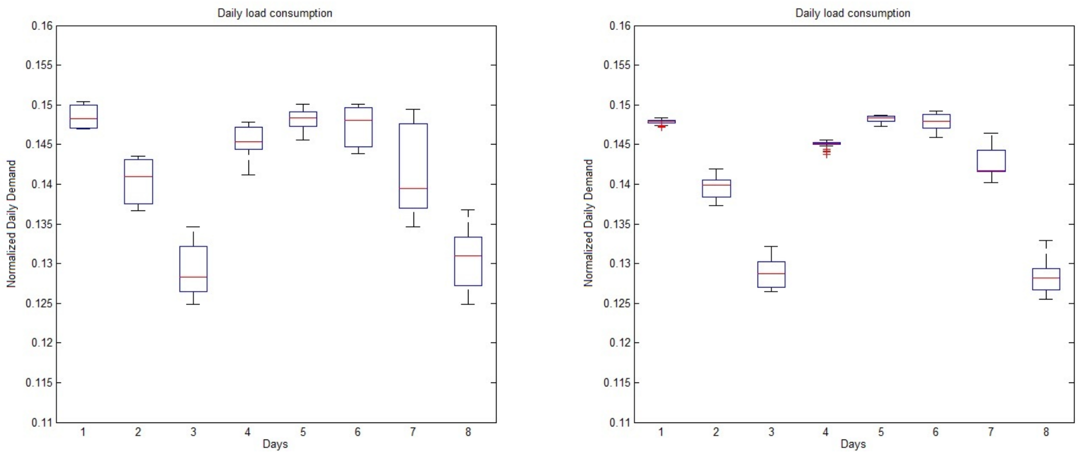
(b) Virtual data of Thursday holidays.

Figure A.15: Box plots represent the evolution of the *densification trick* using ITLMS algorithm on group of Thursday holidays.



(a) Real data of Friday holidays (1). (b) Virtual data of Friday holidays (1).

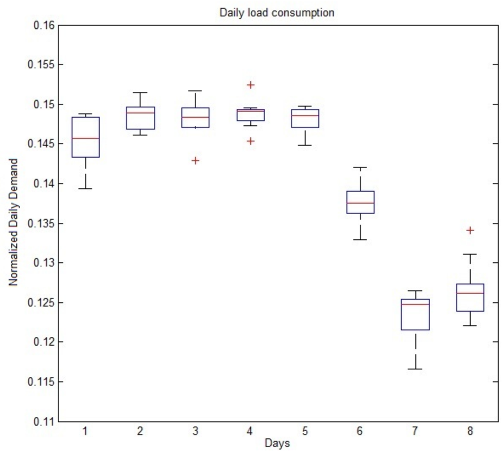
Figure A.16: Box plots represent the evolution of the *densification trick* using ITLMS algorithm on group of Friday holidays (1).



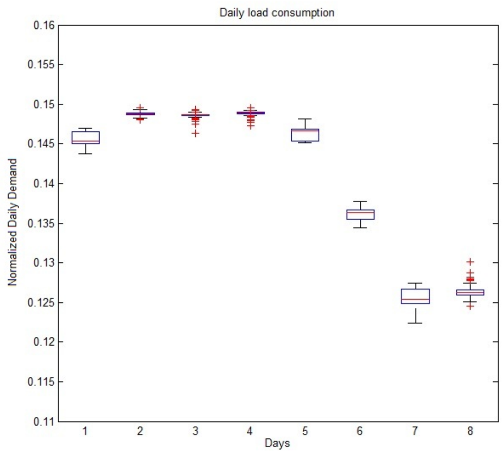
(a) Real data of Friday holidays (2). (b) Virtual data of Friday holidays (2).

Figure A.17: Box plots represent the evolution of the *densification trick* using ITLMS algorithm on group of Friday holidays (2).

A.2.2 Approach 2

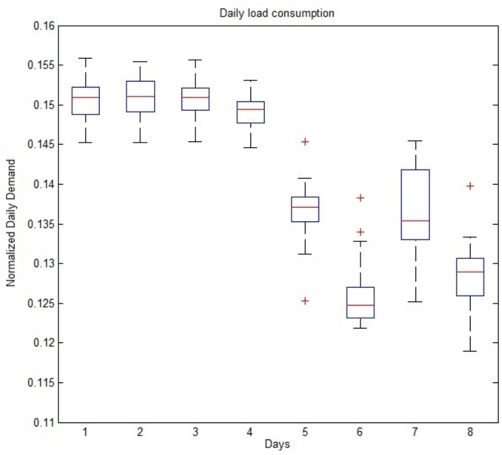


(a) Real data of Monday holidays.

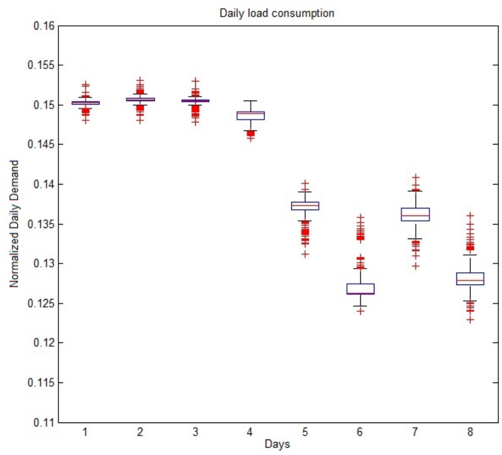


(b) Virtual data of Monday holidays.

Figure A.18: Box plots represent the evolution of the *densification trick* using ITLMS algorithm on group of Monday holidays.

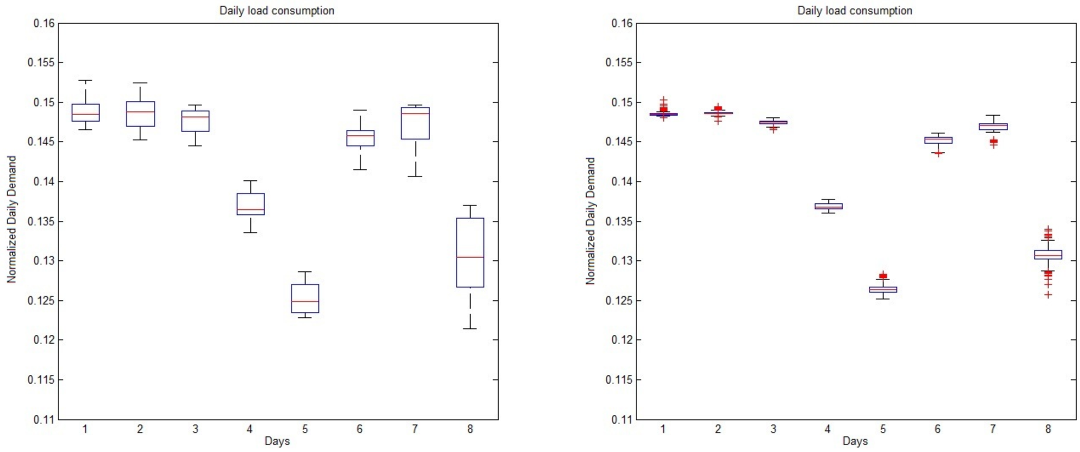


(a) Real data of Tuesday holidays.



(b) Virtual data of Tuesday holidays.

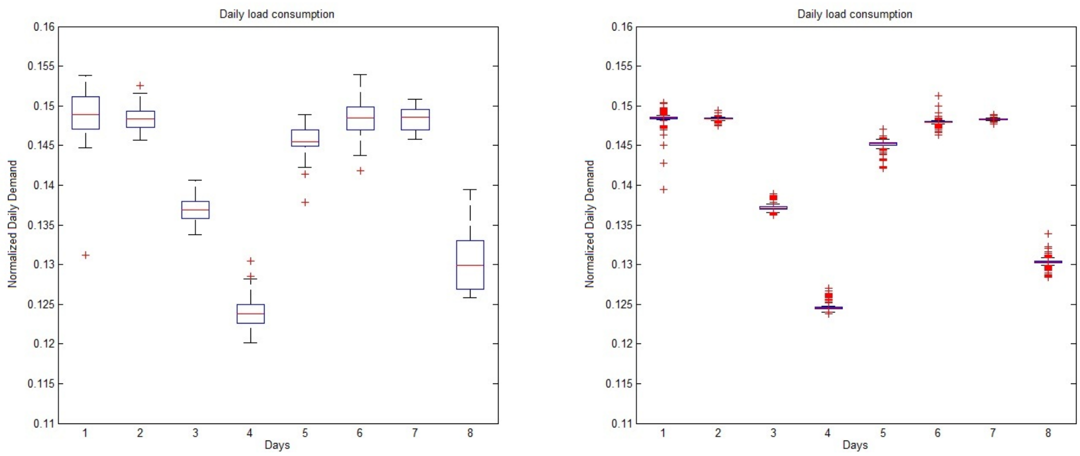
Figure A.19: Box plots represent the evolution of the *densification trick* using ITLMS algorithm on group of Tuesday holidays.



(a) Real data of Wednesday holidays.

(b) Virtual data of Wednesday holidays.

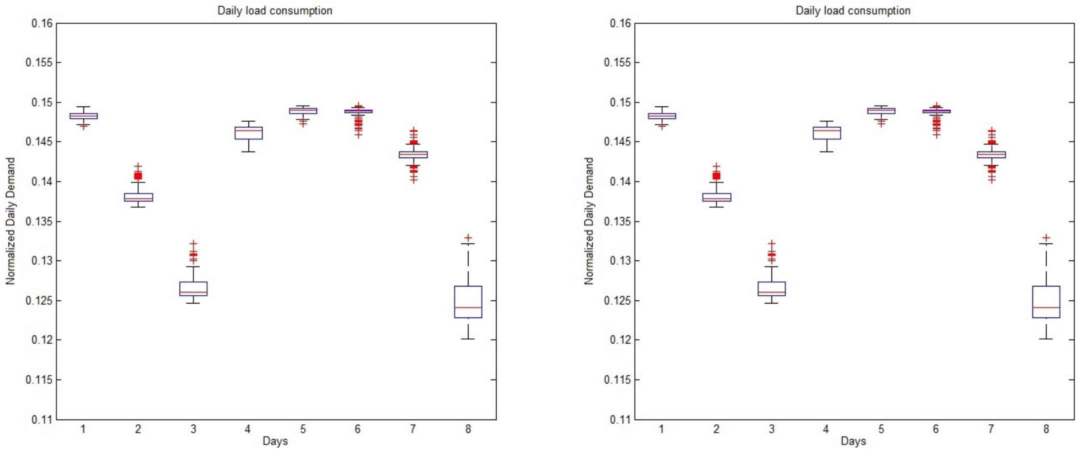
Figure A.20: Box plots represent the evolution of the *densification trick* using ITLMS algorithm on group of Wednesday holidays.



(a) Real data of Thursday holidays.

(b) Virtual data of Thursday holidays.

Figure A.21: Box plots represent the evolution of the *densification trick* using ITLMS algorithm on group of Thursday holidays.



(a) Real data of Friday holidays.

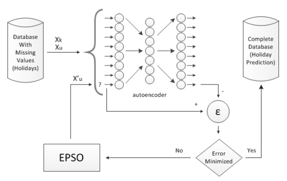
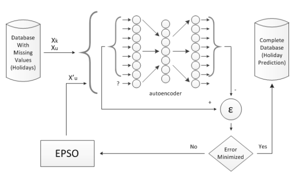
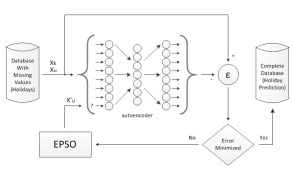
(b) Virtual data of Friday holidays.

Figure A.22: Box plots represent the evolution of the *densification trick* using ITLMS algorithm on group of Friday holidays (1).

Appendix B

Prediction results

Table B.1: Forecasting models, complete description.

<table>
<tr><th>Model</th><th>Optimization search model</th><th>Days preceding the holiday</th><th>Training functions</th></tr>
<tr><td>1</td><td rowspan="6">
</td><td rowspan="2">4</td><td>trainlm</td></tr>
<tr><td>2</td><td>trainbr</td></tr>
<tr><td>3</td><td rowspan="2">6</td><td>trainlm</td></tr>
<tr><td>4</td><td>trainbr</td></tr>
<tr><td>5</td><td rowspan="2">7</td><td>trainlm</td></tr>
<tr><td>6</td><td>trainbr</td></tr>
<tr><td>7</td><td rowspan="6">
</td><td rowspan="2">4</td><td>trainlm</td></tr>
<tr><td>8</td><td>trainbr</td></tr>
<tr><td>9</td><td rowspan="2">6</td><td>trainlm</td></tr>
<tr><td>10</td><td>trainbr</td></tr>
<tr><td>11</td><td rowspan="2">7</td><td>trainlm</td></tr>
<tr><td>12</td><td>trainbr</td></tr>
<tr><td>13</td><td rowspan="6">
</td><td rowspan="2">4</td><td>trainlm</td></tr>
<tr><td>14</td><td>trainbr</td></tr>
<tr><td>15</td><td rowspan="2">6</td><td>trainlm</td></tr>
<tr><td>16</td><td>trainbr</td></tr>
<tr><td>17</td><td rowspan="2">7</td><td>trainlm</td></tr>
<tr><td>18</td><td>trainbr</td></tr>
</table>

B.0.3 Stage 2

Table B.2: Prediction results summary of forecasting models with 7 days preceding the holiday (Monday).

Model	std	MSE	MAE	NMSE	NMAE	MAPE
5	0.134%	8.3E-06	0.217%	0.238%	1.719%	1.694%
6	0.330%	8.2E-07	0.061%	0.023%	0.486%	0.490%
11	0.334%	3.9E-06	0.134%	0.112%	1.063%	1.063%
12	0.350%	2.2E-08	0.010%	0.001%	0.082%	0.083%
17	0.337%	1.6E-06	0.105%	0.047%	0.833%	0.833%
18	0.349%	2.0E-08	0.007%	0.001%	0.053%	0.054%

Table B.3: Prediction results summary of forecasting models with 7 days preceding the holiday (Tuesday).

Model	std	MSE	MAE	NMSE	NMAE	MAPE
5	0.278%	1.5E-05	0.293%	0.366%	2.304%	2.323%
6	0.379%	9.9E-06	0.204%	0.249%	1.608%	1.610%
11	0.464%	1.3E-05	0.243%	0.335%	1.915%	1.902%
12	0.391%	1.4E-06	0.096%	0.036%	0.756%	0.757%
17	0.415%	1.5E-05	0.269%	0.382%	2.120%	2.111%
18	0.375%	1.3E-06	0.073%	0.033%	0.571%	0.569%

Table B.4: Prediction results summary of forecasting models with 7 days preceding the holiday (Wednesday).

Model	std	MSE	MAE	NMSE	NMAE	MAPE
5	0.669%	4.7E-05	0.467%	0.950%	3.572%	3.566%
6	0.454%	2.3E-06	0.071%	0.046%	0.547%	0.554%
11	0.589%	1.4E-05	0.301%	0.23%	2.307%	2.333%
12	0.480%	9.4E-07	0.063%	0.019%	0.483%	0.478%
17	0.404%	1.5E-05	0.308%	0.314%	2.356%	2.384%
18	0.466%	1.6E-07	0.031%	0.003%	0.234%	0.234%

Table B.5: Prediction results summary of forecasting models with 7 days preceding the holiday (Thursday).

Model	std	MSE	MAE	NMSE	NMAE	MAPE
5	0.138%	1.3E-05	0.298%	0.350%	2.282%	2.282%
6	0.277%	8.6E-06	0.221%	0.238%	1.712%	1.712%
11	0.453%	1.1E-05	0.288%	0.308%	2.208%	2.209%
12	0.306%	1.7E-06	0.090%	0.048%	0.691%	0.681%
17	0.373%	7.3E-06	0.233%	0.202%	1.788%	1.786%
18	0.312%	1.1E-06	0.071%	0.031%	0.544%	0.540%

Table B.6: Prediction results summary of forecasting models with 7 days preceding the holiday (Friday (1)).

Model	std	MSE	MAE	NMSE	NMAE	MAPE
5	0.246%	8.9E-09	0.007%	0.0004%	0.058%	0.058%
6	0.237%	8.9E-07	0.036%	0.036%	0.296%	0.299%
11	0.570%	2.5E-05	0.412%	1.007%	3.414%	3.405%
12	0.251%	1.9E-08	0.008%	0.001%	0.066%	0.067%
17	0.279%	4.8E-06	0.154%	0.196%	1.274%	1.257%
18	0.243%	3.4E-09	0.003%	0.0001%	0.027%	0.028%

Table B.7: Prediction results summary of forecasting models with 7 days preceding the holiday (Friday (2)).

Model	std	MSE	MAE	NMSE	NMAE	MAPE
5	0.379%	2.4E-05	0.422%	0.545%	3.226%	3.275%
6	0.442%	1.0E-09	0.003%	2.3E-05%	0.021%	0.021%
11	0.690%	4.6E-05	0.565%	1.043%	4.322%	4.349%
12	0.446%	3.0E-10	0.001%	6.7E-06%	0.008%	0.007%
17	0.639%	1.2E-05	0.312%	0.259%	2.388%	2.387%
18	0.444%	2.0E-10	0.001%	4.4E-06%	0.007%	0.007%

Appendix C

Publications

- Paper to be submitted in a Periodical at Power System

Load Forecasting on Special Days Using ITL Mean Shift and Autoassociative Neural Networks

Daniel Sá, Vladimiro Miranda, *Fellow*, *IEEE* and Jean Sumaili, *Fellow*, *IEEE*

***Abstract*—In this paper, a new concept of load forecasting on special days is presented. Using the Information Theoretical Learning Mean Shift algorithm in a process of densification (*densification trick*) of a scarce data set, resulting in the criation of virtual data to train an Autoassociative Neural Network, allows the use of all real data for validation purposes. The main objective is resolve the problem of not enough amount existence of historical information to represent the special days, such as holidays. This approach is based on Autoassociative Neural Networks as a missing data estimator, in which will be considered the special days as the missing data. An example with daily energy consumption real data from a Brazilian distribution utility illustrates this forecasting model.**

***Index Terms*—Information Theoretic Learning, Mean shift, Autoassociative Neural Networks, load forecasting.**

I. Introduction

NOWADAYS with the deregulation of the power system, requirement of higher efficiency and establishment of new standards on environmental preservation, were introduced harder constraints on the planning, management and control of the power system [1].

Commercial success of the energy companies depends on the ability to submit competitive bids, and improvements in forecasting the load can lead to substantial increases in trading profits.

The quest for top-quality forecasting involves a broad variety of investigation fields, including several areas of engineering, economy, meteorology, and others. The practical details of each particular load forecasting implementation differ from case to case, depending on the objectives. In this paper the repercussion with special days will be taken into account.

Holidays are special days that have a high influence in the load demand curve. In the data set, it is observed that load demand is lower on holidays than on normal days. Moreover, the load demand curve is not only affected on holidays, but also on days located before and after holidays.

According to [2], there are two types of special days, fixed by weekday and fixed by date. A special day fixed by weekday occurs always at the same weekday but its date varies (e.g. Easter). Special days fixed by date fall always at the same time of year (e.g. Christmas). However, it may occur during the weekend in one year and in the middle of the week in the next year.

Load forecasting for holidays is a challenging task once only a small number of recent historical data is available, compared with what is available for normal weekdays. Consequently, average load forecasting errors for the holidays are much higher than those for normal days. Besides, these kinds of events may change the general forecasting operations, channeling the performance to unacceptable levels.

So far, many studies of the load forecasting on special days have been made and the majority are based on neural networks(NN) techniques. Despite the success of these methods none of them has been able to solve the problem of the lack of historical information. Therefore, in this paper, the resolution of this problem is based in the recent work of Sumaili, Miranda et al. [3], where was proposed a new method to solve the problem of the lack of historical data in special days. They were inspired by the results of the Information Theoretical Learning Mean Shift algorithm aplied in a process denoted *densification trick* successfully applied in a problem of incipient fault diagnosis in power transformers [4], where scarce data on failures existed.

Thus, the ITLMS algorithm was used to populate, with virtual data, a scarce set related to daily energy consumption in special days. This allows the proper training of neuronal networks with the virtual data, reserving all the scarce real data for validation purposes. The networks are then used to predict consumption in special days. An example with real data from a Brazilian distribution utility was used in order to illustrate the technique and the same database was used in this work.

Consulting the relevant literature have been observed great results for Autoencoders used as recognition machine [4], [5], with this powerful tool can be estimated missing data in a database. Considered the special days as a missing data, will be evaluated if this tool can adequately predict these kind of days.

In the end will be compared the achieved results with Autoencoders and the results of the work [3] (with a simple Feedforward Neuronal Network), evaluating which of them is the best.

II. Forecasting model

The forecasting model applyed on special days, was inspired in the results of Autoencoders together with optimization methods (e.g. Evolutionary Particle Swam Optimisation (EPSO)) used in the estimation of missing data in a database [4], [5].

Therefore, in this paper the forecasting will be performed by an autoencoder. Autoencoders, are feedforward neural networks with a middle hidden layer that intends to reconstruct the output equal the input, the size of the output layer is always the same as the size of the input layer.

One interesting property of autoencoders is that may be used as a recognition machine. If a new input vector provides different characteristics from the global pattern of the data used for training, the error between the output and input tends to be high, since the result does not match the input.

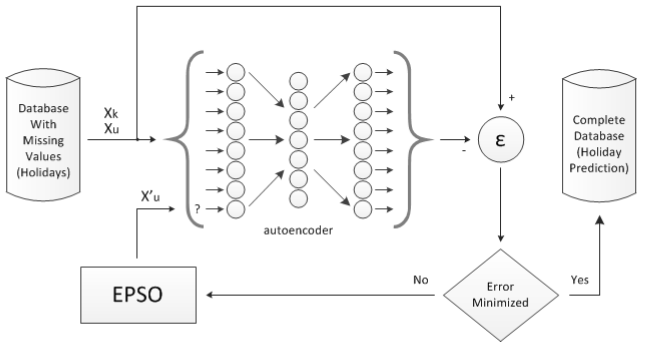


Figure 1. Autoencoder and EPSO - Constrained Search forecasting model.

The constrained search model consists of an optimization algorithm searches for the input values that minimize the input/output error on all the signals. The EPSO metaheuristic is used to estimate the missing values by optimizing the objective function of minimization of the MSE. The complete vector combining the estimated and the observed values is fed into the autoencoder as input. In the figure 1, X_k and X_u represent the known variables and the unknown / missing variables, respectively.

It is important to note that the last neurons in each input and output configuration correspond to the special day, the other inputs / outputs correspond to the days preceding the holiday. Therefore, the load forecasting will be based on the daily energy consumption of the seven days preceding the holiday.

III. Case study: Daily energy consumption forecasting on special days

Allying the densification trick using ITLMS algorithm and Autoencoders used as recognition machine is intended perform the prediction on special days. The following sections describe each phase of this new approach.

A. Data treatment

The historical data which were taken into account in this paper are the same that were applied in [3], the real data from a Brazilian distribution utility. The historical data refer to about 10 years of consumption (from January 2002 to September 2012)

The treatment of the historical data was achieved taking into account the following criteria:

- The forecasting will be based on the daily energy consumption of the seven days preceding the holiday. Weekends will not be analysed, nor consecutive holidays with frequency inferior to eight days;
- The demand of the same special days are dissimilar each year due to the system load growth/decline trend. If this growth is ignored, the general shapes of same days become similar. Therefore, the load forecasting will be performed based on historical data of holidays with the same behavior;
- As mentioned earlier, to solve the problem of the lack of historical data on special days the ITLMS algorithm will be used to make the densification of data set.

The first step to data treatment of the holidays and their previous days is to make a normalization (with respect to the consumption of the previous week) in all of them in order to obtain their similarity. Then, ITLMS algorithm was used to understand the similarity between the patterns. Therefore, it was possible to identify distinct patterns for special days, and cluster them in similar classes.

With setting $\lambda = 0.9$ in a first approach and $\lambda = 0.1$ in a second approach in ITLMS parameters, the identification of thirteen and ten different modes for each approach was possible. The patterns converging to a common mode were grouped in individual clusters. It was thus possible to form clusters corresponding to the five days of the week (from Monday to Friday) and others groups with the remaining outliers that were not taken into consideration in this study.

As stated before, the *densification trick* using ITLMS algorithm will be applied as a way of resolving the problem of lack of historical data, insufficient to an autoencoder training practice. The training set is composed of only virtual points, keeping the totality of the real data to be used in the validation phase. This largely increases the robustness of the validation procedure and the confidence in the results it will provide.

In table I the full description of the database obtained is shown. The number of patterns obtained on database using the properly correction and the ITLMS for classification, and the number of patterns of virtual data created by densification of the real data. The different values of virtual data can be justified with the number of the original real data and the number of iterations needed by the mean shift algorithm to converge to a single mode. In this paper was taken into account two different approachs:

- Approach A: 21 iterations, generating 21 virtual data for each real data;
- Approach B: 59 iterations, generating 59 virtual data for each real data).

Table I
Database complete description.

Approach A ($\lambda = 0.9$) / Approach B ($\lambda = 0.1$)

Special Day	**Real Data**		**Virtual Data**	
	A	B	A	B
Monday	11	11	231	649
Tuesday	19	22	399	1298
Wednesday	14	14	294	826
Thursday	22	22	462	1298
Friday	10	19	210	1121
Friday	6	—	126	—

The reason of the database considering two different groups of special days which occur on Friday in the approach 1 is because these holidays, even falling on same weekday, have a different weekly behavior and the settings given to ITLMS led to consider these days in different clusters.

B. Prediction results

1) The influence of the consideration of two different approachs of densification of data: As the clusters of real data are not equal for the two approaches, this comparison only was made to the three clusters with the same real data. The results achieved were:

Table II
PREDICTION RESULTS SUMMARY. COMPARISON BETWEEN TWO DIFFERENT APPROACHS OF DENSIFICATION OF DATA (A1 AND A2).

Special day	Real days	std		NMAE	
		A	B	A	B
Monday	11	0.349%	0.349%	0.053%	0.053%
Wednesday	14	0.466%	0.463%	0.234%	0.254%
Thursday	22	0.312%	0.312%	0.544%	0.686%

It is possible conclude that this two approaches are equivalents. For the Mondays predictions the same forecasting indicators were obtained. In the other two cluster the results were also very close.

2) Comparison with prediction results given in [3]: The results achieved in the paper [3] will be compared with the results that were given by the model found in this work, considering, logically, the same clusters of data, data sets of the Approach B of the densification of data performed in this paper.

In order to provide a fuller understanding of the differences between this two forecasting methods:

Table III
CHARACTERISTICS OF TWO DIFFERENT FORECASTING METHODS. PROPOSED IN THIS WORK AND PROPOSED IN THE PAPER [3].

Based on Autoencoders	Based on Feedforward NN [3]
Training functions	
Bayesian Regulation Bp.	Simple Backpropagation
Forecasting Model	
Autoencoder 8-7-8, based on missing data estimation, optimization performed by the EPSO in order to predict the special day (missing data) taking in consideration the seven days that precede the holiday.	Simple Feedforward NN 7-3-1, after the proper NN training, the daily consumption forecasting is made considering the seven inputs as the seven days that procede the holidays and the output the holiday predicted.

Table IV
PREDICTION RESULTS SUMMARY. COMPARISON WITH THE RESULTS ACHIEVED IN [3]

Special day	Real days	std	std [3]	NMAE	NMAE [3]
Monday	11	0.35%	1.76%	0.05%	1.85%
Tuesday	22	0.45%	2.50%	0.91%	2.82%
Wednesday	14	0.46%	1.90%	0.25%	3.41%
Thursday	22	0.31%	1.66%	0.69%	2.55%
Friday	19	0.50%	2.46%	1.08%	3.92%

The Normalized Mean Absolute Error (NMAE) varies from 0.05% (Monday) to 1.08% (Friday) for the model with Autoencoder, while the results in [3] were in the order of 1.85% (Monday) to 3.92% (Friday). The variation range of the corresponding standard deviation is from 0.31% to 0.50% better than in [3] with 1.66% to 2.50%.

As can be demonstrated, with this new forecasting model into analysis, better results were achieved.

An improvement of std prediction parameter in the order of 76% (Wednesday) to 82% (Tuesday) was achieved, and for NMAE in the order of 68% (Tuesday) to 97% (Monday).

To all results was verified that the best results were achieved to the clusters with less days in real data. It is simple verify that this occur because with less data is more simple to find the correct pattern while for a cluster with more patterns becomes more difficult to interpolate the correct pattern.

IV. CONCLUSION

This paper demonstrates the advantage of using the Information Theoretic Learning Mean Shift algorithm, in the form of a *desnsification trick*. With this tool became possible the neuronal networks training even when faced with scarce data sets, the problem of special days. The ITLMS can be used to identify distinct clusters in the load data, associating in a easy way holidays that occurs in distinct days of the week and then allow the virtual data collected representing these specific clusters.

It has been proven that the use of the virtual data as training set can be applied as if they were training with real data.

The alliance of this powerful tool with an Auto Associative Neural Network, demonstrated to be a robust model in the load forecasting on special days.

This Autoencoder based on missing data estimation, use an optimization performed by the metaheuristic EPSO in order to predict the special day (missing data) taking in consideration the days that precede the holiday.

The high accuracy achieved by this method confirms that this tools can bring improvements in the performance of the load forecasting methodologies, specially, on days with occurence of scarce historical data to represent their behavior.

Through the obtained results, it can be concluded that Autoencoders based on missing data estimation gives better results than a simple Feedforward Neural Network.

This topology could be an important step in load forecasting on special days and even on normal days.

REFERENCES

[1] J. Fidalgo and J. Lopes, "Load forecasting performance enhancement when facing anomalous events," *Power Systems, IEEE Transactions on*, vol. 20, no. 1, pp. 408–415, 2005.

[2] A. Laukkanen, "The use of special day information in a demand forecasting model for nordic power market," 2004.

[3] J. Sumaili, V. Miranda, L. Rego, Á. Santana, and C. Francês, "A densification trick using mean shift to allow demand forecasting in special days with scarce data," *17th International Conference on Intelligent System Applications to Power Systems*, pp. 1–5, Tokyo, Japan, 2013.

[4] V. Miranda, A. Castro, and S. Lima, "Diagnosing faults in power transformers with autoassociative neural networks and mean shift," *Power Delivery, IEEE Transactions on*, vol. 27, no. 3, pp. 1350–1357, 2012.

[5] V. Miranda, J. Krstulovic, H. Keko, C. Moreira, and J. Pereira, "Reconstructing missing data in state estimation with autoencoders," *Power Systems, IEEE Transactions on*, vol. 27, no. 2, pp. 604–611, 2012.

References

[1] Sudhir Rao, Allan de Medeiros Martins, and J. C. Príncipe. Mean shift: An information theoretic perspective. *Pattern Recogn. Lett.*, 30(3):222–230, February 2009.

[2] Simon Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 1998.

[3] V. Miranda. Evolutionary algorithms with particle swarm movements. In *Intelligent Systems Application to Power Systems, 2005. Proceedings of the 13th International Conference on*, pages 6–21, 2005.

[4] J. Sumaili, V. Miranda, L. Rego, Á. Santana, and C. Francês. A densification trick using mean shift to allow demand forecasting in special days with scarce data. *17th International Conference on Intelligent System Applications to Power Systems*, pages 1–5, Tokyo, Japan, 2013.

[5] *Neural Network Toolbox User's Guide*.

[6] V. Miranda. Redes neuronais - treino por retropropagação. In *Texto de apoio à disciplina de Controlo Difuso e Redes Neuronais no 5º ano da LEEC, FEUP*, pages 207–212, Porto, 2007.

[7] J.N. Fidalgo and J.A.P. Lopes. Load forecasting performance enhancement when facing anomalous events. *Power Systems, IEEE Transactions on*, 20(1):408–415, 2005.

[8] M. Farhadi and M. Farshad. A fuzzy inference self-organizing-map based model for short term load forecasting. In *Electrical Power Distribution Networks (EPDC), 2012 Proceedings of 17th Conference on*, pages 1–9, 2012.

[9] Chin Yen Tee, J.B. Cardell, and G.W. Ellis. Short-term load forecasting using artificial neural networks. In *North American Power Symposium (NAPS), 2009*, pages 1–6, 2009.

[10] Kyung-Bin Song, Young-Sik Baek, Dug Hun Hong, and Gilsoo Jang. Short-term load forecasting for the holidays using fuzzy linear regression method. In *Power Engineering Society General Meeting, 2005. IEEE*, pages 1338 Vol. 2–, 2005.

[11] Pauli Murto. Neural network models for short-term load forecasting. Master's thesis, Helsinki University of Technology, 1998.

[12] Kieran Richard Godden. Electric load forecasting for holiday periods. Master's thesis, Faculty of Science at the Rand Afrikaans University, 1997.

[13] Ching-Lai Hor, S.J. Watson, and S. Majithia. Analyzing the impact of weather variables on monthly electricity demand. *Power Systems, IEEE Transactions on*, 20(4):2078–2085, 2005.

[14] S. Ruzic, A. Vuckovic, and N. Nikolic. Weather sensitive method for short term load forecasting in electric power utility of serbia. *Power Systems, IEEE Transactions on*, 18(4):1581–1586, 2003.

[15] Antti Laukkanen. The use of special day information in a demand forecasting model for nordic power market, 2004.

[16] G. Gross and F.D. Galiana. Short-term load forecasting. *Proceedings of the IEEE*, 75(12):1558–1573, 1987.

[17] H.S. Hippert, C.E. Pedreira, and R.C. Souza. Neural networks for short-term load forecasting: a review and evaluation. *Power Systems, IEEE Transactions on*, 16(1):44–55, 2001.

[18] Yige Zhao, P.B. Luh, C. Bomgardner, and G.H. Beerel. Short-term load forecasting: Multi-level wavelet neural networks with holiday corrections. In *Power Energy Society General Meeting, 2009. PES '09. IEEE*, pages 1–7, 2009.

[19] L.A.D. de Luca, C.M. de Oliveira, and R.S. Wazlawick. Load behavior changes after holidays on thursdays. In *Computational Science and Engineering Workshops, 2008. CSE-WORKSHOPS '08. 11th IEEE International Conference on*, pages 101–106, 2008.

[20] Manoj Kumar. Short-term load forecasting using artificial neural network techniques. Master's thesis, National Institute of Technology Rourkela, 2009.

[21] Wesin Ribeiro Alves. Modelos para previsão de carga a curto prazo através de redes neurais artificiais com treinamento baseado na teoria da informação. Master's thesis, Universidade Federal do Pará Instituto de Tecnologia, 2011.

[22] G. Chicco, Roberto Napoli, and Federico Piglione. Load pattern clustering for short-term load forecasting of anomalous days. In *Power Tech Proceedings, 2001 IEEE Porto*, volume 2, pages 6 pp. vol.2–, 2001.

[23] R. Lamedica, A. Prudenzi, M. Sforna, M. Caciotta, and V.O. Cencellli. A neural network based technique for short-term forecasting of anomalous load periods. *Power Systems, IEEE Transactions on*, 11(4):1749–1756, 1996.

[24] S. Ahmmed, M.A.A. Khan, M.K. Hasan, A.Y. Saber, M.N. Huda, and M.Z. Rahman. Stlf using neural networks and fuzzy for anomalous load scenarios - a case study for hajj. In *Electrical and Computer Engineering (ICECE), 2010 International Conference on*, pages 722–725, 2010.

[25] Dou Quansheng, Pan Guanyu, Shi Zhongzhi, and Yang Bin. Knowledge extraction model for power load characteristics of special days and extreme weather. In *Intelligent Computing and Intelligent Systems, 2009. ICIS 2009. IEEE International Conference on*, volume 1, pages 496–499, 2009.

[26] Qia Ding, Hui Zhang, Tao Huang, and Junyi Zhang. A holiday short term load forecasting considering weather information. In *Power Engineering Conference, 2005. IPEC 2005. The 7th International*, pages 1–61, 2005.

[27] Kwang-Ho Kim, Hyoung sun Youn, and Yong-Cheol Kang. Short-term load forecasting for special days in anomalous load conditions using neural networks and fuzzy inference method. *Power Systems, IEEE Transactions on*, 15(2):559–565, 2000.

[28] M. Farhadi and S. M. Moghaddas-Tafreshi. A novel model for short term load forecasting of iran power network by using kohonen neural networks. In *Industrial Electronics, 2006 IEEE International Symposium on*, volume 3, pages 1726–1731, 2006.

[29] Young-Min Wi, Sung-Kwan Joo, and Kyung-Bin Song. Holiday load forecasting using fuzzy polynomial regression with weather feature selection and adjustment. *Power Systems, IEEE Transactions on*, 27(2):596–603, 2012.

[30] L.A.D. de Luca. Previsao de carga em sistemas de potencia durante feriados prolongados: Efeito do feriado na quinta-feira sobre a carga da sexta-feira. Master's thesis, Universidade Federal de Santa Catarina, 2008.

[31] I. Aquino, C. Perez, J. K. Chavez, and S. Oporto. Daily load forecasting using quick propagation neural network with a special holiday encoding. In *Neural Networks, 2007. IJCNN 2007. International Joint Conference on*, pages 1935–1940, 2007.

[32] Yuan-Yih Hsu and Chien-Chuen Yang. Design of artificial neural networks for short-term load forecasting. i. self-organising feature maps for day type identification. *Generation, Transmission and Distribution, IEE Proceedings C*, 138(5):407–413, 1991.

[33] D. Srinivasan, C.S. Chang, and A.C. Liew. Demand forecasting using fuzzy neural computation, with special emphasis on weekend and public holiday forecasting. *Power Systems, IEEE Transactions on*, 10(4):1897–1903, 1995.

[34] R. Barzamini, M.-B. Menhaj, A. Khosravi, and S. H. Kamalvand. Short term load forecasting for iran national power system and its regions using multi layer perceptron and fuzzy inference systems. In *Neural Networks, 2005. IJCNN '05. Proceedings. 2005 IEEE International Joint Conference on*, volume 4, pages 2619–2624 vol. 4, 2005.

[35] N. Mahdavi, M.-B. Menhaj, and S. Barghinia. Short-term load forecasting for special days using bayesian neural networks. In *Power Systems Conference and Exposition, 2006. PSCE '06. 2006 IEEE PES*, pages 1518–1522, 2006.

[36] S. Barghinia, S. Kamankesh, N. Mahdavi, A. H. Vahabie, and A.A. Gorji. A combination method for short term load forecasting used in iran electricity market by neurofuzzy, bayesian and finding similar days methods. In *Electricity Market, 2008. EEM 2008. 5th International Conference on European*, pages 1–6, 2008.

[37] Qingqing Mu, Yonggang Wu, Xiaoqiang Pan, Liangyi Huang, and Xian Li. Short-term load forecasting using improved similar days method. In *Power and Energy Engineering Conference (APPEEC), 2010 Asia-Pacific*, pages 1–4, 2010.

[38] Ying Chen, P.B. Luh, and S.J. Rourke. Short-term load forecasting: Similar day-based wavelet neural networks. In *Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on*, pages 3353–3358, 2008.

[39] Ying Chen, P.B. Luh, Che Guan, Yige Zhao, L.D. Michel, M.A. Coolbeth, P.B. Friedland, and S.J. Rourke. Short-term load forecasting: Similar day-based wavelet neural networks. *Power Systems, IEEE Transactions on*, 25(1):322–330, 2010.

[40] D.C. Park, M.A. El-Sharkawi, II Marks, R.J., L.E. Atlas, and M.J. Damborg. Electric load forecasting using an artificial neural network. *Power Systems, IEEE Transactions on*, 6(2):442–449, 1991.

[41] Yun Lu, Xin Lin, and Weifu Qi. The method of short-term load forecasting based on the rbf neural network. In *Electricity Distribution, 2005. CIRED 2005. 18th International Conference and Exhibition on*, pages 1–4, 2005.

[42] D. Srinivasan, A.C. Liew, and C.S. Chang. Forecasting daily load curves using a hybrid fuzzy-neural approach. *Generation, Transmission and Distribution, IEE Proceedings-*, 141(6):561–567, 1994.

[43] S.S. Sharif and J.H. Taylor. Real-time load forecasting by artificial neural networks. In *Power Engineering Society Summer Meeting, 2000. IEEE*, volume 1, pages 496–501 vol. 1, 2000.

[44] M.A. Aboul-Magd and E.E.-D.E.-S. Ahmed. An artificial neural network model for electrical daily peak load forecasting with an adjustment for holidays. In *Power Engineering, 2001. LESCOPE '01. 2001 Large Engineering Systems Conference on*, pages 105–113, 2001.

[45] A.P. Alves da Silva, U. P. Rodrigues, A.J.R. Reis, and L.S. Moulin. Neurodem-a neural network based short term demand forecaster. In *Power Tech Proceedings, 2001 IEEE Porto*, volume 2, pages 6 pp. vol.2–, 2001.

[46] S. Barghinia, P. Ansarimehr, H. Habibi, and N. Vafadar. Short term load forecasting of iran national power system using artificial neural network. In *Power Tech Proceedings, 2001 IEEE Porto*, volume 3, pages 5 pp. vol.3–, 2001.

[47] J.W. Taylor and R. Buizza. Neural network load forecasting with weather ensemble predictions. *Power Systems, IEEE Transactions on*, 17(3):626–632, 2002.

[48] T. W S Chow and C.T. Leung. Neural network based short-term load forecasting using weather compensation. *Power Systems, IEEE Transactions on*, 11(4):1736–1742, 1996.

[49] A. Piras, A. Germond, B. Buchenel, K. Imhof, and Y. Jaccard. Heterogeneous artificial neural network for short term electrical load forecasting. *Power Systems, IEEE Transactions on*, 11(1):397–402, 1996.

[50] A. Khotanzad, R. Afkhami-Rohani, T.-L. Lu, A. Abaye, M. Davis, and D.J. Maratukulam. Annstlf-a neural-network-based electric load forecasting system. *Neural Networks, IEEE Transactions on*, 8(4):835–846, 1997.

[51] H. Yoo and R.L. Pimmely. Short term load forecasting using a self-supervised adaptive neural network. *Power Systems, IEEE Transactions on*, 14(2):779–784, 1999.

[52] Ku-Long Ho, Yuan-Yih Hsu, Chuan-Fu Chen, Tzong-En Lee, Chih-Chien Liang, Tsau-Shin Lai, and Kung-Keng Chen. Short term load forecasting of taiwan power system using a knowledge-based expert system. *Power Systems, IEEE Transactions on*, 5(4):1214–1221, 1990.

[53] C.N. Lu, H.-T. Wu, and S. Vemuri. Neural network based short term load forecasting. *Power Systems, IEEE Transactions on*, 8(1):336–342, 1993.

[54] H. B. Gooi, C.Y. Teo, L. Chin, S.Y. Ang, and E. K. Khor. Adaptive short-term load forecasting using artificial neural networks. In *TENCON '93. Proceedings. Computer, Communication, Control and Power Engineering.1993 IEEE Region 10 Conference on*, number 0, pages 787–790 vol.2, 1993.

[55] Hui-Feng Shi and Yan-Xia Lu. Bayesian neural networks for short term load forecasting. In *Wavelet Analysis and Pattern Recognition, 2009. ICWAPR 2009. International Conference on*, pages 160–165, 2009.

[56] Hui-Feng Shi and Yanxia Lu. Short-term load forecasting based on bayesian neural networks learned by hybrid monte carlo method. In *Machine Learning and Cybernetics (ICMLC), 2010 International Conference on*, volume 3, pages 1494–1499, 2010.

[57] E.H. Tito, G. Zaverucha, M. Vellasco, and M. Pacheco. Bayesian neural networks for electric load forecasting. In *Neural Information Processing, 1999. Proceedings. ICONIP '99. 6th International Conference on*, volume 1, pages 407–411 vol.1, 1999.

[58] Yuan Ning, Yufeng Liu, and Qiang Ji. Bayesian - bp neural network based short-term load forecasting for power system. In *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on*, volume 2, pages V2–89–V2–93, 2010.

[59] A. Baniamerian, M. Asadi, and E. Yavari. Recurrent wavelet network with new initialization and its application on short-term load forecasting. In *Computer Modeling and Simulation, 2009. EMS '09. Third UKSim European Symposium on*, pages 379–383, 2009.

[60] S.S. Rao and B. Kumthekar. Recurrent wavelet networks. In *Neural Networks, 1994. IEEE World Congress on Computational Intelligence., 1994 IEEE International Conference on*, volume 5, pages 3143–3147 vol.5, 1994.

[61] Weijian Ren, Zhenghui Zhang, Yubo Duan, Qiong Wang, and Hongli Dong. An adaptive diagonal recurrent wavelet neural network based on compact wavelet frame and its application. In *Control and Automation, 2005. ICCA '05. International Conference on*, volume 1, pages 599–604 Vol. 1, 2005.

[62] S.M. Kelo and S.V. Dudul. Short-term load prediction with a special emphasis on weather compensation using a novel committee of wavelet recurrent neural networks and regression methods. In *Power Electronics, Drives and Energy Systems (PEDES) 2010 Power India, 2010 Joint International Conference on*, pages 1–6, 2010.

[63] S. Uejima H. Tanaka and K. Asai. Linear regression analysis with fuzzy model. In *IEEE Trans. Syst. Man Cybern*, pages vol. 12, pp. 1291–1294, Dec. 1982.

[64] H. Tanaka and J. Watada. Possibilistic linear systems and their application to linear regression model. In *Fuzzy Sets and Syst.*, pages vol. 27, pp. 275–289, 1988.

[65] J. Nazarko and W. Zalewski. The fuzzy regression approach to peak load estimation in power distribution systems. *Power Systems, IEEE Transactions on*, 14(3):809–814, 1999.

[66] J. Nazarko and W. Zalewski. An application of the fuzzy regression analysis to the electrical load estimation. In *Electrotechnical Conference, 1996. MELECON '96., 8th Mediterranean*, volume 3, pages 1563–1566 vol.3, 1996.

[67] S.E. Papadakis, J.B. Theocharis, S. J. Kiartzis, and A.G. Bakirtzis. A novel approach to short-term load forecasting using fuzzy neural networks. *Power Systems, IEEE Transactions on*, 13(2):480–492, 1998.

[68] B. Ye, N. N. Yan, C.X. Guo, and Y.J. Cao. Identification of fuzzy model for short-term load forecasting using evolutionary programming and orthogonal least squares. In *Power Engineering Society General Meeting, 2006. IEEE*, pages 8 pp.–, 2006.

[69] P.K. Dash, S. Dash, and S. Rahman. A fuzzy adaptive correction scheme for short term load forecasting using fuzzy layered neural network. In *Neural Networks to Power Systems, 1993. ANNPS '93., Proceedings of the Second International Forum on Applications of*, pages 432–437, 1993.

[70] P.K. Dash, A.C. Liew, and S. Rahman. Fuzzy neural network and fuzzy expert system for load forecasting. *Generation, Transmission and Distribution, IEE Proceedings-*, 143(1):106–114, 1996.

[71] Ma-WenXiao, Bai-XiaoMin, and Mu-LianShun. Short-term load forecasting with artificial neural network and fuzzy logic. In *Power System Technology, 2002. Proceedings. PowerCon 2002. International Conference on*, volume 2, pages 1101–1104 vol.2, 2002.

[72] P.K. Dash, G. Ramakrishna, A.C. Liew, and S. Rahman. Fuzzy neural networks for time-series forecasting of electric load. *Generation, Transmission and Distribution, IEE Proceedings-*, 142(5):535–544, 1995.

[73] Kwang-Ho Kim, Jong-Keun Park, Kab-Ju Hwang, and Sung-Hak Kim. Implementation of hybrid short-term load forecasting system using artificial neural networks and fuzzy expert systems. *Power Systems, IEEE Transactions on*, 10(3):1534–1539, 1995.

[74] C. Jaipradidtham. Next day load demand forecasting of future in electrical power generation on distribution networks using adaptive neuro-fuzzy inference. In *Power and Energy Conference, 2006. PECon '06. IEEE International*, pages 64–67, 2006.

[75] V. Miranda, A.R.G. Castro, and S. Lima. Diagnosing faults in power transformers with autoassociative neural networks and mean shift. *Power Delivery, IEEE Transactions on*, 27(3):1350–1357, 2012.

[76] Sudhir Rao, Weifeng Liu, J.C. Principe, and A. de Medeiros Martins. Information theoretic mean shift algorithm. In *Machine Learning for Signal Processing, 2006. Proceedings of the 2006 16th IEEE Signal Processing Society Workshop on*, pages 155–160, 2006.

[77] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *Information Theory, IEEE Transactions on*, 21(1):32–40, 1975.

[78] Emanuel Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):pp. 1065–1076, 1962.

[79] Yizong Cheng. Mean shift, mode seeking, and clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(8):790–799, 1995.

[80] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619, 2002.

[81] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010.

[82] A. Pooransingh, C.-A. Radix, and A. Kokaram. The path assigned mean shift algorithm: A new fast mean shift implementation for colour image segmentation. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 597–600, 2008.

[83] Zhi-Qiang Wen and Zi xing Cai. Mean shift algorithm and its application in tracking of objects. In *Machine Learning and Cybernetics, 2006 International Conference on*, pages 4024–4028, 2006.

[84] K.A. Shah, H.K. Kapadia, V.A. Shah, and M.N. Shah. Application of mean-shift algorithm for license plate localization. In *Engineering (NUiCONE), 2011 Nirma University International Conference on*, pages 1–5, 2011.

[85] Pengfei Li, Shaoru Wang, and Junfeng Jing. The segmentation in textile printing image based on mean shift. In *Computer-Aided Industrial Design Conceptual Design, 2009. CAID CD 2009. IEEE 10th International Conference on*, pages 1528–1532, 2009.

[86] A. Renyi. Some fundamental questions of information theory. In *Selected Papers of Alfred Renyi, Akademia Kiado, Budapest*, volume 2, page 526–552, 1976.

[87] J. C. Principe, N. R. Euliano, and W. C. Lefebvre. *Neural and adaptive systems: fundamentals through simulations*. Wiley New York, 2000.

[88] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. In *Science*, pages vol. 313, no. 5786, pp. 504–507, Jul. 2006.

[89] Terence D. Sanger. Optimal unsupervised learning in a single-layer linear feedforward neural network, 1989.

[90] I. T. Jolliffe. *Principal Component Analysis*. Springer, second edition, October 2002.

[91] Nathalie Japkowicz, Stephen Jose Hanson, and Mark A. Gluck. Nonlinear autoassociation is not equivalent to pca. *Neural Comput.*, 12(3):531–545, March 2000.

[92] V. Miranda, J. Krstulovic, H. Keko, C. Moreira, and J. Pereira. Reconstructing missing data in state estimation with autoencoders. *Power Systems, IEEE Transactions on*, 27(2):604–611, 2012.

[93] Mussa Abdella and T. Marwala. The use of genetic algorithms and neural networks to approximate missing data in database. In *Computational Cybernetics, 2005. ICCC 2005. IEEE 3rd International Conference on*, pages 207–212, 2005.

[94] Fulufhelo Vincent Nelwamondo, Dan Golding, and Tshilidzi Marwala. A dynamic programming approach to missing data estimation using neural networks. *Inf. Sci.*, 237:49–58, 2013.

[95] Mohamed S. Nelwamondo, F. V. and T. Marwala. Missing data: A comparison of neural networks and expectation maximization techniques. *Current Science*, 93:pp1514, December 2007.

[96] S. Mohamed and T. Marwala. Neural network based techniques for estimating missing data in databases. *The 16th Annual Symposium of the Pattern Recognition Association of South Africa, Langebaan, South Africa*, pages pp. 27–32, 2005.

[97] Geoffrey Hinton and Ruslan Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504 – 507, 2006.

[98] P. Munro G. W. Cottrell and D. Zipser. Learning internal representations from gray-scale images: An example of extensional programming. *in Proc. 9th Annu. Conf. Cognitive Science Society, Seattle,WA*, 1987.

[99] M.K. Fleming and G.W. Cottrell. Categorization of faces using unsupervised feature extraction. In *Neural Networks, 1990., 1990 IJCNN International Joint Conference on*, pages 65–70 vol.2, 1990.

[100] B. Golomb, T. Sejnowski, and Howard Hughes. Sex recognition from faces using neural networks. In *Applications of Neural Networks*, pages 71–92. Editor), Kluwer Academic Publishers, 1995.

[101] S. Narayanan, II Marks, R. J., J.L. Vian, J. J. Choi, M. A. El-Sharkawi, and B.B. Thompson. Set constraint discovery: missing sensor data restoration using autoassociative regression machines. In *Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on*, volume 3, pages 2872–2877, 2002.

[102] B.B. Thompson, R.J. Marks, and M.A. El-Sharkawi. On the contractive nature of autoencoders: application to missing sensor restoration. In *Neural Networks, 2003. Proceedings of the International Joint Conference on*, volume 4, pages 3011–3016 vol.4, 2003.

[103] Wei Qiao, Zhi Gao, Ronald G. Harley, and Ganesh K. Venayagamoorthy. Robust neuro-identification of nonlinear plants in electric power systems with missing sensor measurements. *Eng. Appl. Artif. Intell.*, 21(4):604–618, June 2008.

[104] Marwala T Leke-Betechuoh B and Tettey T. Autoencoder networks for hiv classification. *Current Science*, 91, No. 11:pp1467–1473, December 2006.

[105] Tshilidzi Marwala Sizwe M. Dhlamini, Fulufhelo V. Nelwamondo. Sensor failure compensation techniques for hv bushing monitoring using evolutionary computing. *Tenerife, Spain*, pages 430–435, December 16-18, 2005.

[106] S. Mohagheghi, G.K. Venayagamoorthy, and R.G. Harley. Optimal wide area controller and state predictor for a power system. *Power Systems, IEEE Transactions on*, 22(2):693–705, 2007.

[107] S. Narayanan, J.L. Vian, J.J. Choi, II Marks, R.J., M.A. El-Sharkawi, and B.B. Thompson. Missing sensor data restoration for vibration sensors on a jet aircraft engine. In *Neural Networks, 2003. Proceedings of the International Joint Conference on*, volume 4, pages 3007–3010 vol.4, 2003.

[108] V. Miranda. Computação evolucionária: uma introdução. Technical report, FEUP, 2005.

[109] Keko H. Miranda, V. and Á. J. Duque. Stochastic star communication topology in evolutionary particle swarms (epso). *International Journal of Computational Intelligence Research (IJCIR)*, 4:105–116, 2008.

[110] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948 vol.4, 1995.

[111] V. Miranda and N. Fonseca. Epso - best-of-two-worlds meta-heuristic applied to power system problems. In *Proceedings of the Evolutionary Computation on 2002. CEC '02. Proceedings of the 2002 Congress - Volume 02*, CEC '02, pages 1080–1085, Washington, DC, USA, 2002. IEEE Computer Society.

[112] Yuhui Shi and Russell C. Eberhart. Parameter selection in particle swarm optimization. In *Proceedings of the 7th International Conference on Evolutionary Programming VII*, EP '98, pages 591–600, London, UK, UK, 1998. Springer-Verlag.

[113] M. Clerc. The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, volume 3, pages –1957 Vol. 3, 1999.

[114] V. Miranda, C. Cerqueira, and C. Monteiro. Training a fis with epso under an entropy criterion for wind power prediction. In *Probabilistic Methods Applied to Power Systems, 2006. PMAPS 2006. International Conference on*, pages 1–8, 2006.

[115] H. Leite, J. Barros, and V. Miranda. Evolutionary algorithm epso helping doubly-fed induction generators in ride-through-fault. In *PowerTech, 2009 IEEE Bucharest*, pages 1–8, 2009.

[116] Naing Win Oo and V. Miranda. Evolving agents in a market simulation platform - a test for distinct meta-heuristics. In *Intelligent Systems Application to Power Systems, 2005. Proceedings of the 13th International Conference on*, pages 6 pp.–, 2005.

[117] V. Miranda and N. Fonseca. Epso-evolutionary particle swarm optimization, a new algorithm with applications in power systems. In *Transmission and Distribution Conference and Exhibition 2002: Asia Pacific. IEEE/PES*, volume 2, pages 745–750 vol.2, 2002.

[118] H. Leite, J. Barros, and V. Miranda. The evolutionary algorithm epso to coordinate directional overcurrent relays. In *Developments in Power System Protection (DPSP 2010). Managing the Change, 10th IET International Conference on*, pages 1–5, 2010.

[119] V. Miranda, L. de Magalhaes Carvalho, M.A. da Rosa, A.M. Leite Da Silva, and C. Singh. Improving power system reliability calculation efficiency with epso variants. *Power Systems, IEEE Transactions on*, 24(4):1772–1779, 2009.

[120] H. Keko, A.J. Duque, and V. Miranda. A multiple scenario security constrained reactive power planning tool using epso. In *Intelligent Systems Applications to Power Systems, 2007. ISAP 2007. International Conference on*, pages 1–6, 2007.

[121] K. Levenberg. A method for the solution of certain nonlinear problems in least squares. *The Quarterly of Applied Mathematics, 2*, pages 164–168, 1944.

[122] Donald W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11(2):431–441, 1963.

[123] Henri Gavin. The levenberg-marquardt method for nonlinear least squares curve-fitting problems. September 28 2011.

[124] M.T. Hagan and M.-B. Menhaj. Training feedforward networks with the marquardt algorithm. *Neural Networks, IEEE Transactions on*, 5(6):989–993, 1994.

[125] David J.C. MacKay. Bayesian interpolation. *Neural Computation*, 4:415–447, 1991.

[126] David J.C. MacKay. A practical bayesian framework for backprop networks. *Neural Computation*, 4:448–472, 1991.

[127] F. Dan Foresee and M.T. Hagan. Gauss-newton approximation to bayesian learning. In *Neural Networks,1997., International Conference on*, volume 3, pages 1930–1935 vol.3, 1997.

[128] Cerqueira C. Monteiro C. A. Miranda, V. Previsão de potênia eólica - treino de sistemas com critérios de entropia.

[129] R. Bessa, V. Miranda, and J. Gama. Wind power forecasting with entropy-based criteria algorithms. In *Probabilistic Methods Applied to Power Systems, 2008. PMAPS '08. Proceedings of the 10th International Conference on*, pages 1–7, 2008.

[130] R.J. Bessa, V. Miranda, and J. Gama. Entropy and correntropy against minimum square error in offline and online three-day ahead wind power forecasting. *Power Systems, IEEE Transactions on*, 24(4):1657–1666, 2009.

[131] R. Linsker. Self-organization in a perceptual network. *Computer*, 21(3):105–117, 1988.