The published version of this accepted manuscript can be found at:
https://doi.org/10.1016/j.asoc.2018.12.028

Parallel Memetic Algorithm for Training Recurrent Neural Networks for the Energy Efficiency Problem

L.G.B. Ruiz[a,*], M.I. Capel[b], M.C. Pegalajar[a]

[a] *Department of Computer Science and Artificial Intelligence, University of Granada.*

[b] *Department of Software Engineering, University of Granada.*

Abstract

In our state-of-the-art study, we improve neural network-based models for predicting energy consumption in buildings by parallelizing the CHC adaptive search algorithm. We compared the sequential implementation of the evolutionary algorithm with the new parallel version to obtain predictors and found that this new version of our software tool halved the execution time of the sequential version. New predictors based on various classes of neural networks have been developed and the obtained results support the validity of the proposed approaches with an average improvement of 75% of the average execution time in relation to previous sequential implementations.

Keywords: energy efficiency; neural networks; time series prediction; evolutionary algorithms; manager-worker parallelization algorithms

[*] Corresponding author. Tel.: +34 958-241-000 / Ext. 20466; e-mail: bacaruiz@ugr.es. At c/ Periodista Daniel Saucedo Aranda, s.n., 18071 Granada, Spain.

1. Introduction

Energy Efficiency (EE) is currently one of the greatest areas of interest for governments since the implementation of building energy-saving policies has a major impact on reducing pollutant emissions and has the greatest potential for delivering significant economic savings. It is also undoubtedly a matter of concern to the international community who must adhere to Directive 2012/27/EU of the European Parliament regarding energy management in terms of compliance with the overall objectives of improving energy efficiency, increasing the use of renewable energy sources, and reducing greenhouse gas emissions [1]. This paper presents a performant solution to the energy consumption prediction problem with time based on a simple parallelization of the CHC adaptive search algorithm [2].

The successful application of models to address the EE issue in buildings or in a distributed location environment typically requires the use of real-time applications [3, 4] if we need to obtain timely, dependable data to feed the mentioned models. Processing this type of information, however, requires the use of fast, accurate techniques so that decisions can be made quickly [5], and energy consumption prediction with time restrictions therefore represents a fundamental problem to be solved in order to achieve exhaustive energy management in buildings.

The problem of consumption forecasting has been explored for building energy management in recent years [6-8]. Nevertheless, these approaches are ad hoc solutions to EE problems and are based on classical techniques which require high computational costs if we want to achieve optimal solutions. Different soft computing paradigms such as artificial neural networks (ANN) [8-10] or evolutionary computation (EC) [6, 11, 12] can be used to find predictable energy management systems that reduce energy waste in buildings: ANN-based systems have proved to be successful for models for energy consumption prediction [9, 10, 13], energy inefficiency diagnosis and fault detection [14, 15]; EC-based ones support multi-objective applications that can be combined with Data Mining (DM). There are several EC applications worth mentioning and these include ones that deal with cost-optimal analysis [16], classification of new electricity customers [17], or selection of the most relevant features [18] and detection of outliers [14, 19, 20].

It is expected that the growing availability of data will soon require innovative tools to face the challenges posed by the variety, volume and velocity of data generation [21]. A sensorized environment generates high speed data streams, which involve the development of fast technology that is capable of processing large amounts of data from all the sensors of the equipment currently used for heating, ventilation, air-conditioning (HVAC) and lighting. We can deploy DM techniques that are strongly supported by parallel computing [22] to greatly reduce the data transfer overload while huge volumes of data are processed. Furthermore, the adaptation of EE-prediction sequential algorithms, which are themselves highly parallelizable, requires the development of designs using powerful parallel techniques which are now available. The application of these methods allows us to obtain valuable benefits in computing time to solve more complex problems by means of parallel approaches [23]. Our proposal for solving such problems is therefore based on estimating the most efficient handling of energy possible by applying ANN and EC techniques and accelerating knowledge extraction and evaluation of data related to energy savings in real time through advanced parallelization techniques.

Although DM techniques require meaningful amounts of data in order to acquire relevant knowledge and reach useful conclusions, they are not generally suited to processing large amounts of data and responding within a reasonable period of time. Addressing these two issues represents a fundamental challenge, especially nowadays, when taking prompt decisions is essential to save costs in terms of energy consumption efficiency [5]. It is essential to optimize energy consumption prediction techniques for processing real-time data in scenarios where the dataflow is constant and permanent [24].

Generally speaking, the main goal of parallel techniques is to detect and exploit the available computational resources in order to make optimal use of them. DM techniques and evolutionary algorithms often present an iterative process which might be a significant opportunity for improvement in terms of time. Finding the optimal subset of code which supplies these requirements is an arduous task and may require a large number of modifications to be made to the sequential algorithm [25].

Our study proposes a modified implementation of the CHC algorithm [26] for optimizing the models used in energy consumption forecasting methods. This optimization algorithm has been widely used in recent studies. There are as many papers as task scheduling policies for providing services to numerous users in cloud environments for solving cloud computing problems [27]. These papers maximize resource utilization and minimize task processing time, or optimize the configuration of a new evolutionary fuzzy k-NN algorithm as in the proposal by Derrac *et al.* [28]. The CHC algorithm is used to establish the model parameters by self-optimization. Within this study domain, articles can be also found that follow the CHC scheme, such as the proposal shown in [29] which can forecast energy consumption from short-term to long-term time series using radial basis function neural networks.

We have used four well-known types of ANN: the non-linear autoregressive neural network (NAR) with exogenous inputs (NARX), the Elman neural network (ENN) and the Long Sort-Term Memory (LSTM) for modelling energy-consumption time series and predicting future consumption using only the historical energy-consumption record. The disadvantage of NAR models being affected by external inputs has been addressed by including NARX models, and the advantage of adding memory to the model by incorporating the ENN and LSTM are included in our study.

The main goal of this paper is to propose a methodology for energy consumption forecasting by making optimal use of existing resources. In addition, since this method provides two essential features (i.e. the good fit of the ANN for time series and the improvement of these models by GA optimization which avoids entrapment in a local minimum), our method therefore enables us to obtain optimal solutions.

The ANN deployed in our study implementation have been fed with raw data with treated missing values extracted from energy consumption meters in buildings on a daily basis. The ANN were trained with real data sets obtained from buildings at our University and the results showed a prediction mean square error of 0.013 in the worst case and 0.0003 in the best. We also showed that different types of NN such as Elman, LSTM can even improve these results.

For NN parallelization, we have deployed a simple parallelization of a map/reduce-like algorithm based on manager-workers which are connected by a crossbar switch on an Intel® Core™ i7-6700 processor (CPU 3.40GHz, 16 GB RAM), which yields an excellent enhancement of the time cost for the four NN used (NAR, NARX, Elman and LSTM) for implementing the model's algorithms.

The paper is structured as follows: Section 2 presents the methodology proposed to obtain a feasible solution to energy consumption time series prediction in buildings, this section also introduces mathematical models of NAR, NARX, Elman and LSTM and its graphical topologies; Section 3 examines the genetic algorithm used to model the time series of one of the energy consumptions for one of the buildings in this study; Section 4 discusses the dataset comprising raw consumption-data for one year from various buildings at our University; Section 5 details the results obtained in the different tests conducted in the study; and finally, Section 6 outlines our conclusions and details some practical implications.

2. Methodology

This section presents the proposed method for energy time series prediction and for minimizing cost over time, enabling full advantage to be taken of available energy resources and innovations to be developed that will provide better results when applied to the use of these resources. The first part of our method deals with data collection and pre-processing. The second part examines the forecast modelling tool. The third step explores genetic optimization with the integration of parallelization techniques. In the final step, the obtained results are validated and analyzed.

Diverse techniques have been employed to solve forecasting problems for many years with different scopes. For instance, in medicine —studies have been carried out to predict and reduce abdominal aortic aneurysm diseases using hemodynamic prediction [30] or to predict drug responses in cancer based on multiple types of genome using Regression Vector Machine [31]—, in marketing —data analysis of data produced by social networks such as Facebook, YouTube, LinkedIn and Twitter to predict influenza epidemic or stock market trends using Self Organizing Fuzzy Neural Networks and Support Vector Machine [32]— or environmental sciences —Jung et al. [33] applied a Genetic Algorithm and a Least Squares Support Vector machine to predict daily building energy consumption and in Deb et al. [34] a complete time series forecasting methods review employed in this subject is done, where other techniques are employed in recent years, such as Grey prediction models and Fuzzy Systems—. There are also recent works that combine time series techniques such as Discrete Wavelet Transform and Empirical Mode Decomposition in order to improve electric load forecasting [35] or even, Deep Learning models, i.e.: Xueheng et al. [36] propose an ensemble Deep Learning model with Empirical Mode Decomposition for load demand prediction, and demonstrate that these models show advantages when prediction horizon increases.

The scientific community has conducted a large number of studies into the problem of energy time series forecasting. Artificial Neural Networks (ANN) have proved promising because of the good/excellent results [2, 37, 38] they yield. In this paper, we use four widely known ANN models: the non-linear autoregressive neural network (NAR) and the non-linear autoregressive neural network with exogenous inputs (NARX), Elman neural network (ENN) and Long Short-Term Memory neural network (LSTM).

The NAR network allows us to model energy-consumption time series most simply. This model is capable of predicting future consumption by using only the historical energy-consumption record. Xian Zhang *et al.* report good performance forecasts for electric vehicle sales in the automobile industry with an NAR neural network [39], although, as they also point out, the main disadvantage of NAR models is that they may be affected by other external factors. As a result, it is necessary to extend ANN-based models in order to be able to integrate more information that can enrich these models [40].

The ENN is a less well known model, however, this neural network introduces a new significant term, crucial when historical information is processed. This is the concept of *memory*. Thus, the ENN's architecture adds a new temporal component to consider previous states in the network to predict the future values of the time series. The ENN has demonstrated excellent performance, especially at the time series problems where past behaviour guides future responses [41, 42] and have proved to be a strong competitor against NAR and NARX models [6].

Finally, due to the increasing interest in Big Data technologies and Deep Learning methods, the LSTM neural network has become very popular over the past few years. The LSTM is the most sophisticated model of all presented here and is also a strong competitor if sufficient data is available. Some studies compare ENN and LSTM architectures achieving very similar results [43], in that study, Mohab et al. show that the ENN is stronger than the authors expected and benefits greatly from their approach. The LSTM models have been also exploited in the energy field and have yielded remarkable outcomes [44, 45].

In this study, a new more efficient algorithm must therefore be implemented to use computing resources in the best possible way. In this context, one classic CHC [26] has been adapted here to improve both ANN accuracy and time-cost.

2.1. NAR and NARX models

Artificial neural network are very powerful, accurate techniques and are currently used for modelling and predicting in various fields. There are recent proposals in medicine which combine the NAR neural network and the autoregressive integrated moving average (ARIMA) to forecast the incidence of tuberculosis [46] or applications of this ANN for predicting incidence tendency of haemorrhagic fever with renal syndrome [47]. In the sphere of finance, these models have been used to forecast stock market returns [48], for fraud detection [49], or even smart card security for public transportation applications based on a novel neural network analysis of cardholder behaviour [50]. In the EE domain, there is a broader array of applications, e.g. simulation-based energy optimization is presented in [51] by applying a web-based parallel genetic algorithm to reduce the computation time for a series of test buildings in Spain. Petri *et al.* present a modular optimization model for reducing energy consumption in large-scale building facilities using ANN [52]. An updated review of time series-based forecasting techniques for building energy consumption can be found in [34].

We should first define the concept of energy consumption before modelling it. To this purpose, energy consumption can be described as a time-series $y(t)$ which represents the energy consumption performed at time t. In many circumstances, the data obtained belong to a fleeting, transient and ephemeral behaviour of the building energy consumption and since this decreases the effectiveness of linear methods, a non-linear approach is therefore recommended. A non-linear autoregressive neural network (NAR) can be modelled using Equation (1), where $\hat{y}$ is the current value of a data series y at time t, modelled by the p past-values of the series. In principle, $h(\cdot)$ is an unknown non-linear function which is approximated by the optimizing process that is carried out to obtain the optimal weights and bias of the network. The error of the network's estimation of the value y at time t [9, 53] is represented by $\epsilon(t)$:

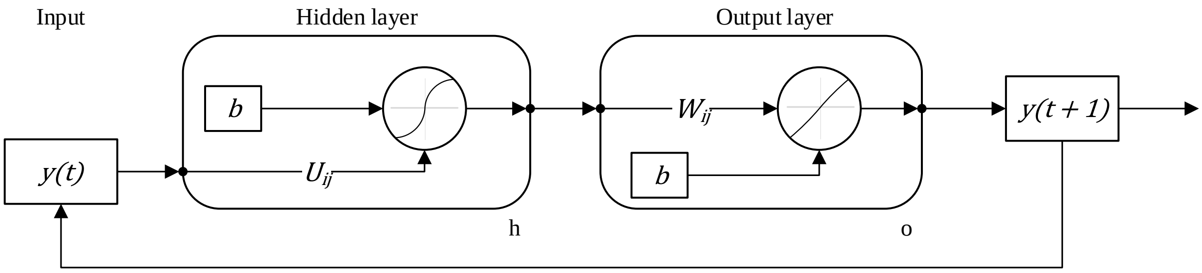
(1)

Similarly, the non-linear autoregressive with exogenous inputs (NARX) is also used when data not only depend on the total amount spent but also on other possible factors of influence. Nagy *et al.* [54] use weather conditions as a model feature, and a more accurate predictor of energy consumption is supplied for conducting the time series modelling process and previous work has proved that this has certain advantages [9]. However, one such advantage of using extra information by dynamic feedback input comes with one main disadvantage: it provides a more complex alternative model, where the uncertainty of the additional data may limit the expected performance of the initial model. The NARX model can be described as the following mathematical function:

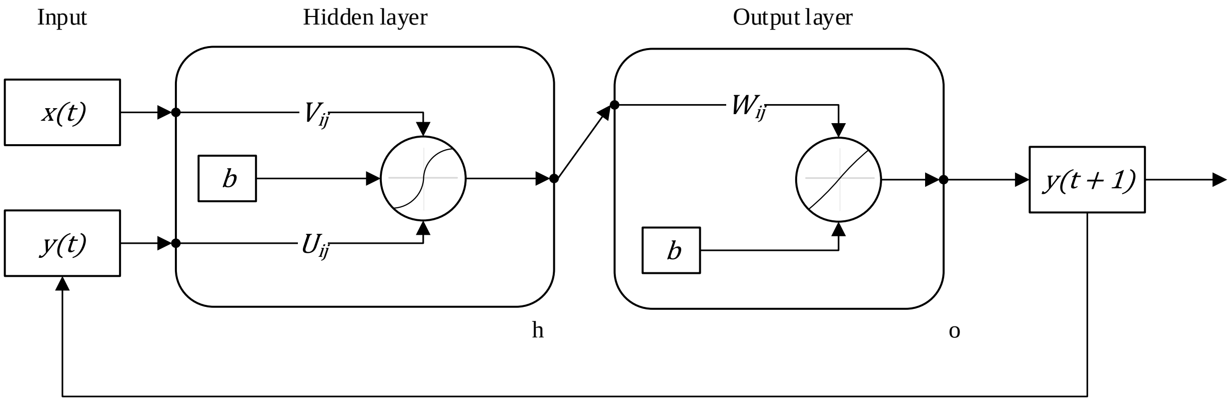
(2)

where $x(t-i)$ is the external time series at time $t-i,\ i \in [1,p]$ and p are the number of past values used. It should be noted that the NAR model given by x is zero. The structure of these two models can be found in Fig. 1. Both models are described by the U and matrices. U_{ij} is the weight between input i and the hidden neuron j, and is the number of neurons in the hidden layer. represents the weight between the hidden neuron i and the output neuron and o defines the number of output neurons in this layer. The last matrix, V, specifies the weights for the connections between the exogenous input and the hidden layer. The inputs and $x(t-q)$ where are the input time series with p past values and the previous values employed to model the future value of the series).

In every case, the parameter is the bias associated with its neuron.



(a)



(b)

Fig. 1. Representing the structure of a non-linear autoregressive neural network (NAR) using one time series as the input and the next value (a) as the output, and with exogenous inputs NARX (b) where an extra time series is included in the input layer.

2.2. Elman Neural Network

The Elman model (ENN) is a type of network with a recurrent topology and was elaborated by Jeffrey Elman [55]. ENN are a satisfactory time series forecasting method and have proved to be a fast, accurate tool for making future predictions in a wide range of scenarios [56-58]. As with the previous ANN, these models can be found in financial time series prediction to forecast the stock market price indexes [41] and have many different applications such as that proposed by Chu *et al.* in [59] which presents an ENN to identify elderly fall signals. In the EE field, an ENN was developed by Kelo and Dudul [42] to predict electrical power load due to temperature variation. A hybrid model is proposed in [60] for short-term load forecasting and the article also includes a genetic algorithm to achieve the optimal ENN structure. A combination of wavelet and this recurrent network is developed by Sami *et al.* to identify the location of energy transmission faults [61].

The energy forecasting problem works with the evolution of data over time and results in a model that is capable of recording previous results, because the consumption normally shows a cyclic behaviour that justifies ENN deployment. This network introduces recurrence to the network through the addition of a set of units called context (or state) to introduce the concept of *memory*. State neurons acquire the input from the previous hidden layer and return the output to the next hidden layer. This recurrent connection allows the ENN to detect and learn time-varying patterns.

The first difference between the ENN and NAR(X) models is the context layer shown in Fig. 2. The *state neuron layer* has the previous values of the hidden nodes obtained previously: at time the output of the hidden neuron will be the input of all the hidden neurons at time t 1 and therefore, at time t, the context units will have the hidden neurons values at time t [62]. Our decision to adopt this kind of neural network is supported by previous studies, where ENN have yielded significant results, thus demonstrating their usefulness and effectiveness [6, 57, 60-63].

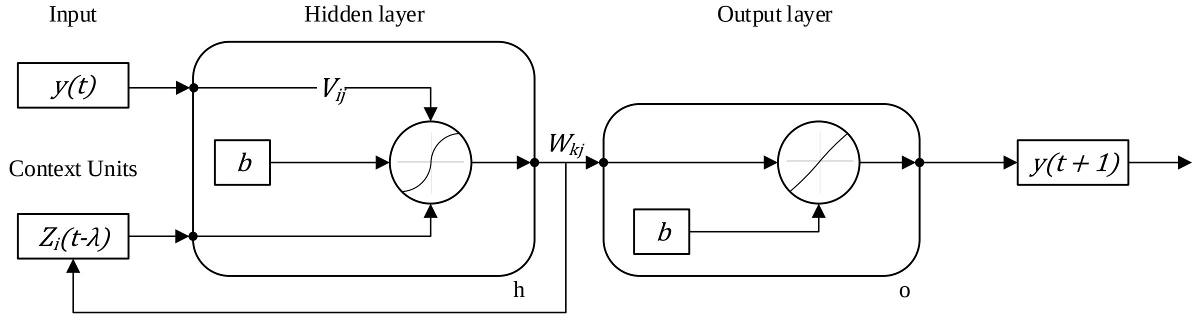


Fig. 2. Representing the topology of the Elman Neural Network.

The output of the ENN $y(t+1)$ is calculated in Equations (3) and (4). The equations detail a more general ENN, with a number of k outputs, the values of index k of the equation above can be further adjusted to fit future applications accordingly, and $y_k(t+1)$ represents the output of neuron k in the last layer. In this study, 1, and G is the activation function of the output layer; is the weight associated with the connection of the hidden neuron j and the neuron k of the output layer; and is the state value corresponding to the neuron r at time t.

(3)

Similarly, is the output of the neuron j in the hidden layer and is calculated as follows:

(4)

where is the weight of the connection between the input neuron j and the hidden neuron i; is the input i at that time and is the activation function of the hidden neurons; I and are the number of neurons in the input and hidden layer, respectively; and is the weight of the connection between the neuron j in the context layer and the neuron r in the hidden layer. In the figure, $Z_j(t-\lambda)$ represents the value of the hidden neuron j at time $t-\lambda$ where and indicates the past values of the hidden neurons stored.

The Levenberg-Marquardt (LM) backpropagation algorithm has been used as the network training function in order to update weight and bias values, according to LM optimization. This is often the fastest algorithm that ensures the best convergence and yields a minimum error in function approximation problems [64].

2.3. Long Short-Term Memory Neural Network

The recurrent neural networks with long short-term memory (LSTM) have recently risen as a powerful and scalable model for diverse learning problems related to sequential data. LSTM, in a similar way to ENN, are effective at learning temporal dependences with the advantage that they do not experience the optimization barriers of the simple recurrent networks [65] and have been employed to solve countless problems. This covers activity recognition —i.e.: Ordóñez and Roggen [66] suggest a Deep Convolutional framework for activity recognition based on convolutional and LSTM recurrent units— handwriting recognition —i.e.: Xiaoqiang et al. propose an innovative recurrent neural method to learn discriminate binary codes, and they use LSTM to learn feature vector by using the convolutional feature map as input for image retrieval [67]—, handwriting generation, language modeling and translation, acoustic modeling of speech, speech synthesis, analysis of audio and video among others [65].

Due to the explosive growth of data in recent years, it is common to find these kind of models with several hidden layers and with a high number of neurons in order to deal with the high complexity and the vast amount of information to be processed. As stated above, the LSTM neural network provides a more complex architecture than ENN, and it is not utilized if little data is available. In this paper, the LSTM architecture chosen for this problem is illustrated in Fig. 3. The main components of the LSTM are a sequence input layer and the LSTM layer. The first layer is the time series data, the second layer is a recurrent layer that enables support for time series and sequence data in the ANN and learns temporal dependencies between time steps of sequence data. Finally, the architecture ends with a fully connected layer which multiplies the input by a weight matrix and then adds a bias vector, and a regression output layer. An important characteristic of the LSTM is that it has been designed to learn to bridge time intervals in excess of 1000 steps even in case of incompressible, noisy input data, without loss of short-time-lag capabilities [68].

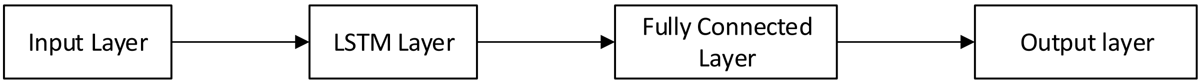


Fig. 3. Representing the topology of the Long Short-Term Memory Neural Network.

LSTM architecture distinguishes itself from the rest by the Memory Cells and the Gate Units. Fig. 4 shows the structure of the cell c_j and its gate units in and out. The self-recurrent connection indicates feedback with a delay of d time steps. The hidden units explicitly manage the flow of information as a function of both the state and input. The state stored in this structure is either deleted by a forget gate or saved indefinitely. Knowledge is thus guaranteed to be transferred over long lapses of time.

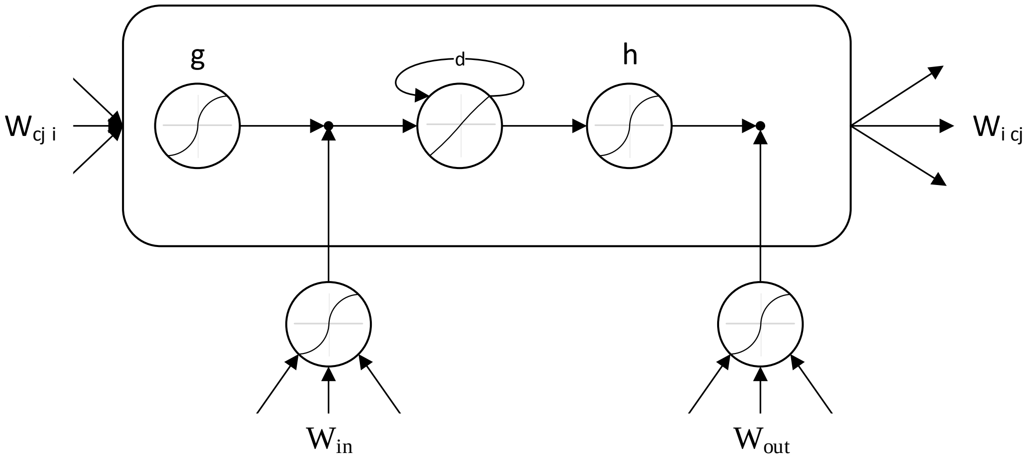


Fig. 4. Basic architecture of memory cell of the LSTM neural network.

In this case, the ADAM algorithm is used for training as the gradient based optimizer, instead of SGD method. The ADAM outperformed SGD in terms of faster convergence and lower error ratios [69]. The main challenge of the training algorithms —in our case: LM for NAR, NARX and Elman networks, and ADAM for LSTM— is that it is typical for solutions to converge to a local minimum. We therefore propose an evolutionary algorithm in order to obtain better outcomes and to optimize the ANN results. The suggested GA is discussed further in the following section. We have selected these four neural networks for testing the validity, adaptive capacity and the reliability of our proposal working with different ANN architectures.

3. Genetic Algorithm

A Genetic Algorithm (GA) is a stochastic optimization method based on the concept of natural evolution. GAs comprise a population of chromosomes (or individuals) and each represents the possible solution to the problem. Each individual has an associated objective value which designates the goodness degree of a solution. Furthermore, the GA has three essential functions: selection, crossover and mutation.

In this paper, the adaptation of the binary "Cross generational elitist selection, Heterogeneous recombination, and Cataclysmic mutation" (CHC) algorithm has been adapted for real-coded solutions [26]. The CHC algorithm finds the optimal neural network weights and biases, and has been adapted to a parallel approach in order to reduce time-cost computation.

The computation time taken to reach a good solution, and of course to improve this solution, is one of the main driving forces behind this study. Maximizing the potential of available resources is an important task and one that is often neglected in many studies and rarely explored as it is in this paper. A large number of publications focus on developing excellent models but do not mention the time cost involved, even though constrained time cost is a common requirement for industry and business.

Nevertheless, various examples of published studies can be found and He and Sun [70] presented their convolutional neural network research to fulfil the requirement of a constrained time budget. They investigate the accuracy of these models under a constrained time cost and design a very fast, accurate architecture that reaches the top-5 error. Lee *et al.* presented an advanced stochastic time-cost trade-off analysis, based on a critical path method guided by a genetic algorithm in order to reduce the computation time, reliability and usability of a previous algorithm. They use the GA for optimization and also to identify the new initial parent chromosomes [71]. A least squares support vector machine to predict building energy consumption improved with real coded GA is used in [33], the purpose of which is to obtain a faster computation speed and greater prediction accuracy. The method performed better in terms of convergence time and iteration economy.

The study presented here exploits the advantages of the CHC algorithm for searching for good solutions and the ANN's disadvantage of falling in a local minimum. Our proposal also benefits from the computational capabilities of the CPU to cope with the high GA time-cost. The proposed parallel GA improves the excessive computation time by distributing the iterative tasks to different workers. We should first explain our coding of the ANN algorithm and so by considering Fig. 1, an individual can be codified as shown in Fig. 5.

w_{11}^{x}	w_{12}^{x}	…	w_{hn}^{x}	w_{11}^{y}	w_{12}^{y}	…	w_{hn}^{y}	w_{11}	w_{12}	…	w_{ho}

Fig. 5. Genetic encoding of a neural network

where n is the number of connexions between the input and the first network layer, h is the number of neurons in the hidden layer and o the number of neurons in the output layer; w_{ij}^{x}

represents the weight between the input neuron j and the hidden neuron i; w_{pq} represents the weight associated with the recurrent connection between the output neuron p, and the hidden neuron q, and represents the weight associated with the connection between the hidden neuron and the output neuron v. When the chromosome acts as NAR neural network architecture, then is not part of the solution. The structure of the assumed algorithm is shown in Fig. 6. This figure includes a flowchart to explain the different steps of the procedure.

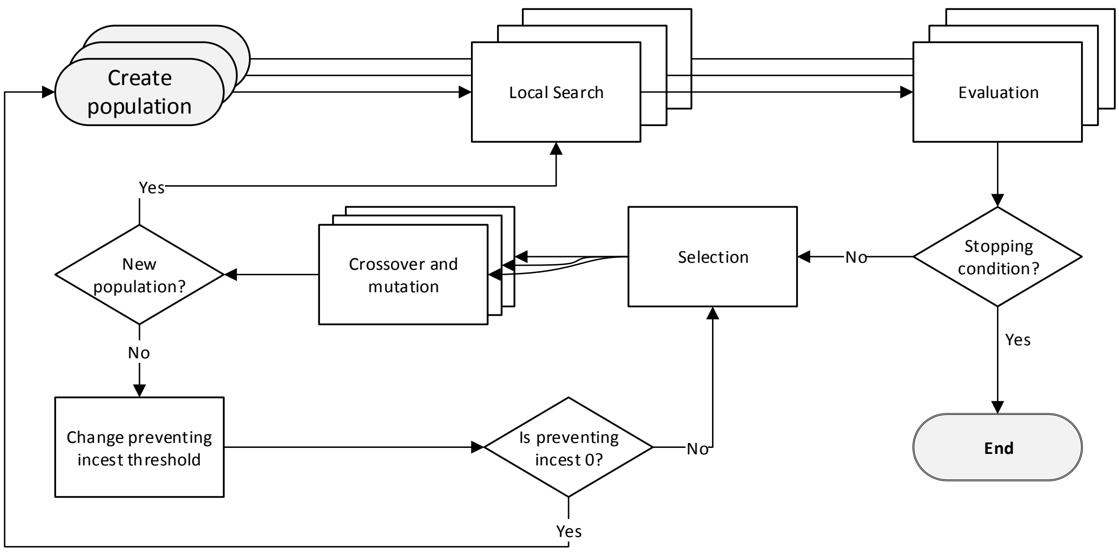


Fig. 6. Flowchart of the proposed genetic algorithm, adapted from Eshelman's CHC.

Each step is described as follows:

1. A totally random, initial population is created. Each gene is initialized in a defined range] and the individuals are created in parallel. Parallelism is performed at the level of the chromosome, illustrated in Fig. 7a.
2. The local search is performed according to the following formula:

(5)

where $D(t)$ is the expected response in the output neurons on the instant time t, $F(\cdot)$ is the activation function of the output neurons, $S(t)$ is a vector containing the hidden neuron state of the network on t, and $X(t)$ represents a vector with input values of the network on t.

The prime aim is to minimize an objective function which depends on the network weights and the expected output. This idea is illustrated by the following equation:

$$G(s) = \sum_{t=0}^{T} (\|D(t) - O(t)\|)^2 = E^T E \tag{6}$$

where $D(t)$ is the expected output on the instant time t, $O(t)$ is the output on the instant t, E refers to a matrix with $Tx1$ values with the output layer error on t, and s is the variable vector to optimize. The parallelism in this step is presented at the level of the individual. According to resources available, a number of individuals are simultaneously optimized with the local search as shown in Fig. 7b.

The hybridization of the evolutionary algorithm and local search method results in the well-known memetic algorithm (MA).

3. The individual objective value is measured using mean square error (MSE) in order to optimize an individual with the local search:

$$MSE = \frac{1}{n-m} \sum_{t=1}^{n} \big(D(t) - O(t)\big)^2 \tag{7}$$

where n is the sample size and m is the number of parameters in the model. This function is parallelized as in the two previous steps. Computer resources are divided up among the individuals so as to avoid idle times which force 100% CPU to be used. This approach is illustrated in Fig. 7a.

4. The individuals are selected using the roulette method [72]. Each individual in the population is assigned a probability of being selected. This probability is proportional to its adjustment, in other words, to its error. The best individuals receive a greater slice of roulette than the worst.

The operator for generating new offspring is the BLX-α crossover operator [73]. A new son $H = (h_1, \dots, h_n)$ is born according to a random number h_i selected in interval $[c_{min} - I \cdot$

$\alpha, c_{max} + I \cdot \alpha]$, where $c_{max} = \max(c_i^1, c_i^2)$, $c_{min} = \min(c_i^1, c_i^2)$, $I = c_{max} - c_{min}$, and c_i^p is the gene i of the parent chromosome p. The parameter α is introduced by the user between $[0, 1]$. This procedure has been parallelized at the gene level as illustrated in Fig. 7b. The new genes of an individual are calculated in tandem. In this phase, this approach has been followed because the crossover is performed if it overcomes a defined probability. Since not all the chromosomes reproduce, this could result in idle times.

5. The mutation operator is responsible for selecting an individual gene and setting a random value between $[G_{min}, G_{max}]$. This step has been parallelized as in the previous phase.
6. The next population shall be constructed by the m best individuals by considering individuals of the previous population and its offspring.
7. The algorithm ends when a number of generations g is achieved.

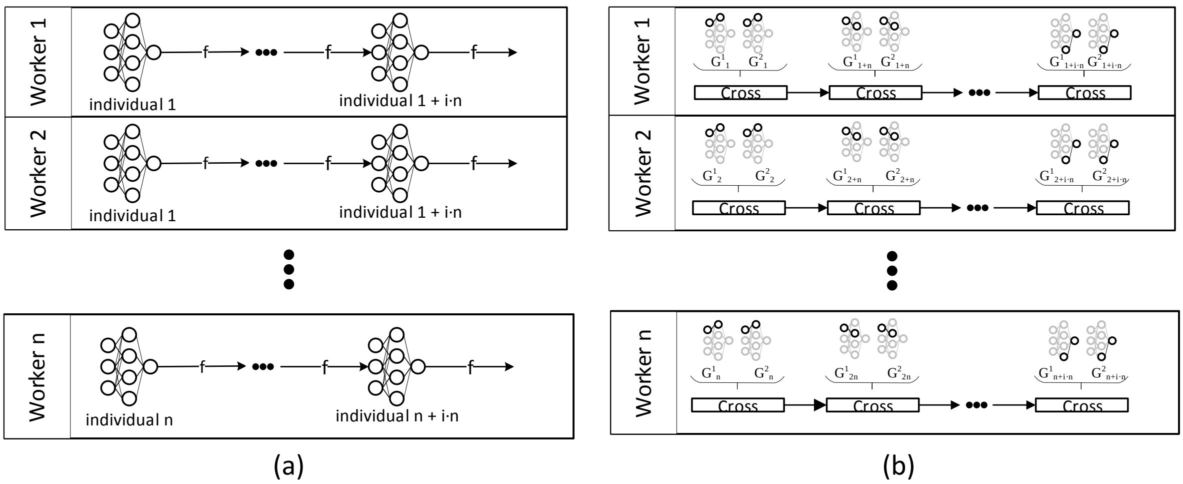


Fig. 7. Parallel distribution of the loads into $\boldsymbol{n}$ workers, $\boldsymbol{f}$ denotes procedures performed on the individual, procedures which include individual creation, local search and objective function (a). Parallel distribution of the chromosome divided into n workers. G_t^p denotes the gene t of the parent p (b).

There is also a diversification strategy to prevent local minimum stagnation so that the algorithm can continue to search the solution space and for this purpose, a re-initialization mechanism is applied. This procedure means that it is necessary to return to Step 1. The criterion established for resorting to this procedure is if there has been no improvement in the results in x generations.

The parallelism depicted in Fig. 7a splits the load into n workers defined as follows:

$$CPU = \{w \tag{8}$$

The total number of models is k:

(9)

These workers are the control process units (CPUs) and each CPU assumes the operations for the individual set C_{W_j}, according to Equation (10):

(10)

where α is the number of assignments given to each worker and calculated as follows:

$$- \tag{11}$$

Similarly, the cross-cutting of chromosome (t genes in length) is carried out by means of the following equation:

(12)

where β is the total number of genes allocated to the available workers:

$$- \tag{13}$$

4. Dataset

In this work, a data set has been collected from a building automation system that records energy-consumption over time. These specific systems usually control energy wastage due to the heating, ventilation, air conditioning and lighting systems of a building. Our study uses data on energy-consumption and weather conditions of the University of Granada (UGR, Granada, Spain). The dataset includes data from two buildings with the same demographic characteristics.

The UGR comprises five campuses: Centro, Cartuja, Fuentenueva, Aynadamar and Ciencias de la Salud, spread over the city of Granada. In total there are 22 colleges, 5 technical engineering schools, 8 training centres and 5 additional centres for culture, sport and general services.

Since current Spanish Data Protection Laws prevent us from specifying the exact location of the buildings and facilities, we numbered them from 1 to 8. Buildings have been selected in light of two representative energy-consumption data for each campus.

Fig. 8 depicts two examples of raw consumption-data for one year. These two consumptions show a linear upward trend since the original data are recorded by the building's energy meter. The raw data shown in the figure therefore represent consumption to date, i.e. the information stored is cumulative consumption. The energy consumed c_t at time t is calculated using the energy price at that moment in time, D_t, and the previous one D_{t-1} as the following equation illustrates:

$$c_t = D_t - D_{t-1} \tag{14}$$

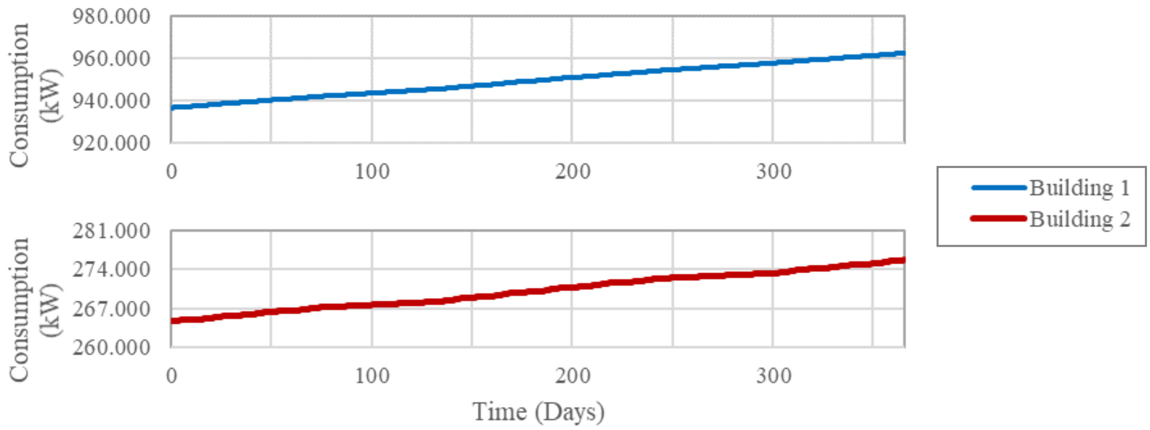


Fig. 8. Meter readings for two examples of raw data consumption over a year.

5. Results

This section presents all of the results obtained in the various tests that have been conducted. Fig. 8 gathered an example of raw data from an energy meter, which periodically accrue the total consumption. It is therefore necessary to summarize and transform the data in order to present them in a usable format. Any incomplete, noisy or unreliable data are also dealt with, and incomplete data have been filled using a linear interpolation imputation method. This method fits a straight line between the endpoints of the gap and enables the missing values to be calculated in a straightforward way by employing the following line equation [74]:

(15)

The value k is calculated as follows:

(16)

where is the interpolant, x symbolizes the time point of the interpolant, and are the coordinates of the starting point of the gap, and indicate the coordinates of the end point of the missing interval.

Fig. 9 depicts two examples of the results captured by the imputation method. It should be noted that linear imputation is the best choice in this problem due to its simplicity and quick computation and because it responds reasonably to the consumption behaviour. Although other imputation methods have been tested (e.g. cubic spline imputation, shape-preserving piecewise cubic interpolation, previous neighbour interpolation, next neighbour interpolation, nearest neighbour interpolation), their behaviour is undesirable and meaningless in this problem. Two such examples of this behaviour are negative consumption or a zero consumption period with a single large spike.

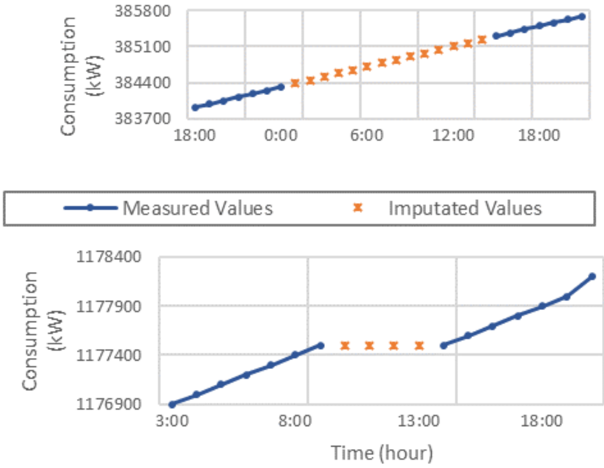


Fig. 9. Two examples of linear imputation values for the energy meters.

The raw data with missing values that have been treated are modified to assemble daily consumption. Energy consumption has also been normalized between 0 and 1 to have the same range of values for each input to the NN, thereby ensuring that the model does not give more weight to the higher range attributes. The data would then be reconstructed as shown in Fig. 10.

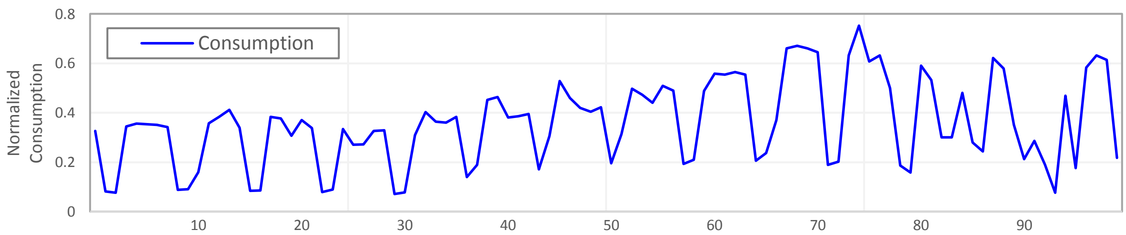


Fig. 10. Daily normalized consumption over a 100-day period.

So that such results may be obtained, the following parameters have been set for the GA: the population size has been set to 25 individuals; the maximum and minimum gene values are 10 and -10, respectively; the stop criteria were established at 100 generations; and the crossover and mutation probability has been set at 90% and 10%. In order to train all the models, the dataset has been randomly split into training and test sets with 70% of the examples allocated to training and 30% to testing. So as to obtain the best parameters for the models, ten tests have been run. Table 1 displays all the errors for each network structure and reports the best number of neurons for each case. It is evident that the NAR and NARX models obtain the best error with 10 neurons, although the optimum is achieved in two consumptions with 8 and 9 neurons, respectively. However, for most buildings, the ENN obtains the optimum solution with 9 neurons. Likewise, the best results of the LSTM network are acquired with 8, 9 and 10 , although if Table 1 is looked closely, we realise that if we were to remove the top three cases (cases 2, 3 and 10), 5 and 7 become the best neuron parameters.

Table 1
Results of experimental time series consumptions: comparison of the three neural networks optimized using the memetic algorithm to display the mean square error

	Neurons									
Model	1	2	3	4	5	6	7	8	9	10
NAR										
Cons. 1	0.0215181	0.0129090	0.0118595	0.0110900	0.0107224	0.0100682	0.0099646	0.0098942	0.0093493	0.0092648
Cons. 2	0.0401715	0.0211694	0.0190867	0.0172446	0.0162044	0.0159827	0.0154314	0.0146710	0.0149999	0.0147922
Cons. 3	0.0367283	0.0205938	0.0168575	0.0155420	0.0146244	0.0142023	0.0132543	0.0128636	0.0128842	0.0122394
Cons. 4	0.0008630	0.0008324	0.0008291	0.0008266	0.0008268	0.0008216	0.0008212	0.0008205	0.0008222	0.0008217
Cons. 5	0.0134595	0.0105834	0.0100412	0.0091431	0.0089837	0.0086461	0.0083847	0.0082914	0.0082652	0.0080677
Cons. 6	0.0279272	0.0132078	0.0101502	0.0089602	0.0083126	0.0079794	0.0073542	0.0069932	0.0067491	0.0064842
Cons. 7	0.0377470	0.0191129	0.0151643	0.0140364	0.0134968	0.0128224	0.0121629	0.0117132	0.0114873	0.0114223
Cons. 8	0.0254187	0.0153050	0.0136017	0.0112555	0.0105364	0.0100957	0.0101201	0.0102848	0.0097331	0.0097186
NARX										
Cons. 1	0.0172769	0.0134489	0.0110335	0.0102405	0.0105713	0.0100784	0.0096436	0.0096312	0.0090996	0.0092894
Cons. 2	0.0351362	0.0206990	0.0192705	0.0180993	0.0180576	0.0168802	0.0158672	0.0155706	0.0144174	0.0146818
Cons. 3	0.0326071	0.0191206	0.0170564	0.0165537	0.0158502	0.0149622	0.0143571	0.0129777	0.0135710	0.0127612
Cons. 4	0.0009212	0.0008938	0.0008991	0.0009013	0.0009001	0.0008833	0.0009095	0.0008910	0.0008932	0.0008797
Cons. 5	0.0104558	0.0084836	0.0079764	0.0073544	0.0073420	0.0070150	0.0069223	0.0066998	0.0065222	0.0064522
Cons. 6	0.0200192	0.0112923	0.0096900	0.0089700	0.0085108	0.0079184	0.0078619	0.0074871	0.0074983	0.0071489
Cons. 7	0.0396858	0.0197610	0.0159657	0.0150626	0.0146919	0.0125092	0.0116735	0.0117385	0.0113365	0.0112290
Cons. 8	0.0187236	0.0130301	0.0106711	0.0099076	0.0097993	0.0093895	0.0092843	0.0092153	0.0090727	0.0088996
Elman										
Cons. 1	0.0181928	0.0080617	0.0071009	0.0065562	0.0064105	0.0057446	0.0053506	0.0039186	0.0030866	0.0036516
Cons. 2	0.0327652	0.0145094	0.0133701	0.0120689	0.0110340	0.0092417	0.0093951	0.0072378	0.0075160	0.0058396
Cons. 3	0.0326860	0.0116246	0.0099089	0.0095616	0.0092085	0.0083863	0.0079345	0.0076725	0.0065537	0.0060289
Cons. 4	0.0008494	0.0007999	0.0007916	0.0006404	0.0007576	0.0006625	0.0003674	0.0005721	0.0006332	0.0007974
Cons. 5	0.0118462	0.0056386	0.0053088	0.0052252	0.0049409	0.0044106	0.0040186	0.0036410	0.0033035	0.0033780
Cons. 6	0.0220548	0.0065676	0.0056810	0.0056224	0.0052802	0.0042862	0.0038940	0.0033211	0.0024795	0.0028352
Cons. 7	0.0331632	0.0108750	0.0094952	0.0093967	0.0088470	0.0085891	0.0076741	0.0071685	0.0062178	0.0068952
Cons. 8	0.0200358	0.0075694	0.0068414	0.0067670	0.0063285	0.0061455	0.0056810	0.0048660	0.0041578	0.0047037
LSTM										
Cons. 1	0.0092897	0.0033385	0.0020597	0.0025722	0.0021547	0.0014317	0.0010614	0.0011855	0.0006317	0.0009949
Cons. 2	0.0421114	0.0129763	0.0084729	0.0081993	0.0050532	0.0041174	0.0031338	0.0032180	0.0033550	0.0027195
Cons. 3	0.0291953	0.0085740	0.0035432	0.0044459	0.0028404	0.0025456	0.0017075	0.0012760	0.0016341	0.0014364
Cons. 4	0.0022123	0.0019221	0.0018247	0.0018727	0.0018015	0.0017007	0.0015570	0.0015211	0.0013734	0.0014758
Cons. 5	0.0191191	0.0102992	0.0082567	0.0065753	0.0049700	0.0046626	0.0048182	0.0052757	0.0042051	0.0037791
Cons. 6	0.0397496	0.0232968	0.0143120	0.0088149	0.0077170	0.0065207	0.0069664	0.0081868	0.0033985	0.0045700
Cons. 7	0.0293439	0.0106638	0.0057946	0.0041042	0.0034333	0.0012890	0.0023552	0.0009120	0.0014171	0.0010371
Cons. 8	0.0232106	0.0067959	0.0054637	0.0031309	0.0036474	0.0024250	0.0023462	0.0020388	0.0013697	0.0016317

The results are summarized in Table 2, in nearly all cases the best are obtained with the LSTM network, nevertheless, there are three cases where the Elman network achieves better outcomes: in every case, Elman and LSTM are well below half the computed MSE values for the other two models. It is also interesting to note that the NAR and NARX networks have a similar error. Although in previous studies, the NARX models with the exogenous input performed best in every case [9], here the MA optimizes the NAR in such a way that it enables a better result to be obtained in the fit of the NARX neural networks in three cases: Consumption 3, 4 and 6. The errors of both models are quite similar.

Table 2
Mean square error performance of the best prediction NAR, NARX, Elman and LSTM networks optimized with the memetic algorithm.

Building	NAR	NARX	Elman	LSTM
Consumption 1	0.0092648	0.0090996	0.0030866	**0.0006317**
Consumption 2	0.0146710	0.0144174	0.0058396	**0.0027195**
Consumption 3	0.0122394	0.0127612	0.0060289	**0.0012760**
Consumption 4	0.0008205	0.0008797	**0.0003674**	0.0013734
Consumption 5	0.0080677	0.0064522	**0.0031310**	0.0037791
Consumption 6	0.0064842	0.0071489	**0.0024795**	0.0033985
Consumption 7	0.0114223	0.0112290	0.0062178	**0.0009120**
Consumption 8	0.0097186	0.0088996	0.0041578	**0.0013697**

One example of the application of our proposal is illustrated in Fig. 11 for case 5, which has the most similar MSE. This illustrates the prediction evolution of the different neural networks performed by the memetic algorithm at various instances of the algorithm, and more specifically, Generations 1, 25, 50, 75 and 100. This graph displays the evolution of MSE performed by the three ANN during the optimization process. It should be noted that the first error obtained has been omitted because of its high value so as not to distract attention from the other results. Fig. 11a shows ANN prediction and the real value of the series in the first generation of the algorithm. It is easily apparent that these models return an almost random prediction because their weights and bias have been randomly initialized, and it is not possible to obtain good results in a single generation. Otherwise, adopting this approach would not be justified.

Nevertheless, Fig. 11b shows how all the models are able to fit the curves more clearly with 25 generations. Fig. 11f supports this assertion because there is an important decrease in the estimated error in every case between generations 40 and 50, after which there is a gradual reduction in the MSE. During the 50th generation, there is improvement in the neural networks and various local peaks have been refined, such as the estimation of the prediction of the consumption at Day 48.

Successive improvements, however, are barely noticeable. Furthermore, from the 50th generation to the end, as the NARX model is not able to improve, the population is reinitialized since the incest threshold has been crossed and no improvement has been found. Something similar occurs with the NAR models which have a softer learning curve than other models. The NAR network was trapped in a local minimum and its population is reset, but in this case, the model achieves a better solution near the 90th generation.

In conclusion, the ENN and LSTM produces very similar results. It is interesting to see how there are considerably wider fluctuations in the ENN in Fig. 11a and these are soon well calibrated. These are not apparent in Fig. 11f because of their high MSE as we explained previously. Another interesting behaviour is illustrated in the same Fig. 11a where LSTM yields the worst prediction, however, it presents a fair view of the trend throughout the whole series. The Fig. 11b shows how the LSTM begins to adjust better and its predictions are in much the same way NAR and NARX models, but soon starts improving and its results are close to the results of the ENN network. In this instance the ENN model produces the best fit in every generation compared to LSTM and also achieves a better forecasting the more generations are performed.

Table 3
Execution time in seconds. Comparison of sequential and parallel memetic algorithm with NAR, NARX, Elman and LSTM models.

	Sequential				Parallel			
Cons.	NAR	NARX	Elman	LSTM	NAR	NARX	Elman	LSTM
1	756	1207	10351	2123	169	242	4714	681
2	835	1582	10330	1990	194	372	4652	557
3	74	1095	10402	2508	166	231	4523	698
4	675	1524	10260	2560	181	353	5084	694
5	555	1040	8908	2480	161	243	4571	718
6	386	1556	5462	2082	117	192	1957	517
7	623	1582	11073	2560	189	351	4631	720
8	574	1129	10937	2504	169	268	4630	646
Mean	644	1339	9715	2351	168	281	4345	654

Finally, Table 3 shows the time cost executions and the computational cost in seconds for every experiment. The table has nine columns: the first identifies the building, and the remaining columns summarize the average execution time breakdown for each test performed. The MA takes 644.29 seconds (≈ 11 minutes) with NAR networks with the sequential version and 168.33 seconds (≈3 minutes) with the parallel process. This represents a time cost improvement of up to 73.87%. It should be noted that the NAR model is the fastest method because its topology is simpler than the NARX and Elman networks. Similarly, the parallel and sequential approaches of the MA with the NARX networks have a time cost of 1339.43 seconds (≈22 minutes) and 281.48 (≈5 minutes), respectively, with a time cost improvement of 78.98%. An unexpected result is obtained with the LSTM and ENN networks, the ENN takes longer to provide the optimal results, and it has an average improvement of 55.28%. Nonetheless, the LSTM achieves in more than half the cases a better error than ENN. On the other hand, LSTM spends far less time to optimize the models, improving time cost by 72.18%. According to the Matlab documentation, the Elman networks are no longer recommended to use, instead they suggest NARX and NAR. This is probably happening because ENN is not optimized in the same way than NAR, NARX and LSTM neural networks.

The codes are executed in Intel® Core™ i7-6700 CPU 3.40GHz, 16 GB RAM memory and Microsoft's Windows 10 (x64) operating system with Matlab R2018a. In accordance with these features, the number of workers has been set to 8.

Fig. 11. Example of the optimization process in 100 generations; the model forecasting for the 50-day period in the first generation (a); optimization achieved in Generation 25 (b), in Generation 50 (c), in Generation 75 (d) and the final generation (e); and the mean square error achieved during the process (f) in each generation.

6. Conclusions

This article introduces a modified implementation of the CHC adaptive search algorithm to improve and optimize energy consumption forecasting models. The novelty of our approach is to show how the CHC algorithm can be modified to produce a parallel memetic energy efficiency prediction proposal in order to satisfactorily improve the algorithm's time cost.

This paper examines the problem of developing technologies to predict future energy consumption in buildings and the ensuing temporary constraints which are crucial for appropriate building energy management. We tested the usefulness of the genetic algorithm to improve solutions, and explored a method to speed up the process by using parallel techniques which have been applied.

We have compared the performance and computational cost of parallel and sequential implementations of the MA to achieve optimal predictors. The parallel algorithm has been found to be computationally much more efficient than the sequential version, with no negative impact on the quality of the solutions. The proposed, properly optimized models are extremely valuable tools for predicting energy consumption, and parallelization of the optimization method provides a 50% reduction in time in the worst case. The predictor models NAR, NARX, ENN and LSTM are successfully developed and the results support the validity of the proposed approach, achieving an average improvement of 75%.

Our experiments highlight the fact that the ENN and LSTM are the most suitable technique for energy consumption prediction. We should also highlight that all of the models shown here provide good results in terms of time-cost, which confirms the importance of our proposal in every test conducted.

Although good results were obtained in the experiments, in the future we want to enhance the model to improve energy management and cost saving. By way of future work, we therefore plan to use feature selection and clustering methods to detect consumption profiles and abnormal consumption, and identify the relationship between supposedly independent consumption periods and peak demand. We also intend to pinpoint groups or patterns of behaviour with certain specific features. The use of a MapReduce approach will improve the scalability of the methodology and this will result in better system efficiency and greater computational capacity so that larger amounts of data can be processed.

By way of conclusion, although this study achieves good results in reasonable execution times, it would be extremely interesting to perform an additional comparative study using differential evolution (DE) approaches. Additionally, incorporating a new randomization-based method and applying decomposition to the time series will probably improve the study results in future work. Furthermore, the development of new deep learning models to deal with this problem will be an interesting alternative to explore for comparison with our method.

7. Acknowledgments

This work has been developed with the support of the Department of Computer Science and Artificial Intelligence of the University of Granada, TIC111, and Project TIN201564776-C3-1-R.

8. Abbreviations

ANN	Artificial Neural Network.
ARIMA	Autoregressive Integrated Moving Average.
CHC	Cross generational elitist selection, Heterogeneous recombination, and Cataclysmic mutation.
DM	Data Mining.
EC	Evolutionary Computation.
EE	Energy Efficiency.
ENN	Elman Neural Network.
GA	Genetic Algorithm.
LM	Levenberg-Marquardt.
LSTM	Long Short-Term Memory.
MA	Memetic Algorithm.
MSE	Mean Square Error.
NAR	Non-linear autoregressive.
NARX	Non-linear autoregressive with external input.
UGR	University of Granada.

9. Bibliography

[1] E.E. Directive, Directive 2012/27/EU of the European Parliament and of the Council of 25 October 2012 on energy efficiency, amending Directives 2009/125/EC and 2010/30/EU and repealing Directives 2004/8/EC and 2006/32, Official Journal, L, 315 (2012) 1-56. [http://ec.europa.eu/energy/en/topics/energy-efficiency]

[2] R. Bhandari, J. Gill, An Artificial Intelligence ATM forecasting system for Hybrid Neural Networks, International Journal of Computer Applications, 133 (2016) 13-16. [http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.734.7389&rep=rep1&type=pdf]

[3] T. Ekwevugbe, N. Brown, V. Pakka, D. Fan, Real-time building occupancy sensing using neural-network based sensor network, in: 2013 7th IEEE International Conference on Digital Ecosystems and Technologies (DEST), 2013, pp. 114-119. [http://dx.doi.org/10.1109/DEST.2013.6611339]

[4] C. Fiori, K. Ahn, H.A. Rakha, Power-based electric vehicle energy consumption model: Model development and validation, Appl Energ, 168 (2016) 257-268. [http://dx.doi.org/10.1016/j.apenergy.2016.01.097]

[5] J. Jackson, Promoting energy efficiency investments with risk management decision tools, Energy Policy, 38 (2010) 3865-3873. [http://dx.doi.org/https://doi.org/10.1016/j.enpol.2010.03.006]

[6] L.G.B. Ruiz, R. Rueda, M.P. Cuéllar, M.C. Pegalajar, Energy consumption forecasting based on Elman neural networks with evolutive optimization, Expert Systems with Applications, 92 (2018) 380-389. [http://dx.doi.org/10.1016/j.eswa.2017.09.059]

[7] D. Masa-Bote, M. Castillo-Cagigal, E. Matallanas, E. Caamaño-Martín, A. Gutiérrez, F. Monasterio-Huelín, J. Jiménez-Leube, Improving photovoltaics grid integration through short time forecasting and self-consumption, Appl Energ, 125 (2014) 103-113. [http://dx.doi.org/10.1016/j.apenergy.2014.03.045]

[8] M. Macarulla, M. Casals, N. Forcada, M. Gangolells, Implementation of predictive control in a commercial building energy management system using neural networks, Energy and Buildings, 151 (2017) 511-519. [http://dx.doi.org/http://dx.doi.org/10.1016/j.enbuild.2017.06.027]

[9] L.G.B. Ruiz, M.P. Cuellar, M. Delgado, M.C. Pegalajar, An Application of Non-Linear Autoregressive Neural Networks to Predict Energy Consumption in Public Buildings, Energies, 9 (2016) 21. [http://dx.doi.org/10.3390/en9090684]

[10] H. Khosravani, M. Castilla, M. Berenguel, A. Ruano, P. Ferreira, A Comparison of Energy Consumption Prediction Models Based on Neural Networks of a Bioclimatic Building, Energies, 9 (2016) 57. [http://www.mdpi.com/1996-1073/9/1/57]

[11] E. Asadi, M.G.d. Silva, C.H. Antunes, L. Dias, L. Glicksman, Multi-objective optimization for building retrofit: A model using genetic algorithm and artificial neural network and an application, Energy and Buildings, 81 (2014) 444-456. [http://dx.doi.org/10.1016/j.enbuild.2014.06.009]

[12] G. Kumar, M.K. Rai, An energy efficient and optimized load balanced localization method using CDS with one-hop neighbourhood and genetic algorithm in WSNs, Journal of Network and Computer Applications, 78 (2017) 73-82. [http://dx.doi.org/10.1016/j.jnca.2016.11.013]

[13] M. Beccali, G. Ciulla, V. Lo Brano, A. Galatioto, M. Bonomolo, Artificial neural network decision support tool for assessment of the energy performance and the refurbishment actions for the non-residential building stock in Southern Italy, Energy, (2017). [http://dx.doi.org/10.1016/j.energy.2017.05.200]

[14] F. Magoulès, H.-x. Zhao, D. Elizondo, Development of an RDP neural network for building energy consumption fault detection and diagnosis, Energy and Buildings, 62 (2013) 133-138. [http://dx.doi.org/10.1016/j.enbuild.2013.02.050]

[15] T. Muhammed, R.A. Shaikh, An analysis of fault detection strategies in wireless sensor networks, Journal of Network and Computer Applications, 78 (2017) 267-287. [http://dx.doi.org/10.1016/j.jnca.2016.10.019]

[16] F. Ascione, N. Bianco, C. De Stasio, G.M. Mauro, G.P. Vanoli, A new methodology for cost-optimal analysis by means of the multi-objective optimization of building energy performance, Energy and Buildings, 88 (2015) 78-90. [http://dx.doi.org/10.1016/j.enbuild.2014.11.058]

[17] J.L. Viegas, S.M. Vieira, R. Melício, V.M.F. Mendes, J.M.C. Sousa, Classification of new electricity customers based on surveys and smart metering data, Energy, 107 (2016) 804-817. [http://dx.doi.org/10.1016/j.energy.2016.04.065]
[18] J.-P. Burochin, B. Vallet, M. Brédif, C. Mallet, T. Brosset, N. Paparoditis, Detecting blind building façades from highly overlapping wide angle aerial imagery, ISPRS Journal of Photogrammetry and Remote Sensing, 96 (2014) 193-209. [http://dx.doi.org/https://doi.org/10.1016/j.isprsjprs.2014.07.011]
[19] F. Duque-Pintor, M. Fernández-Gómez, A. Troncoso, F. Martínez-Álvarez, A New Methodology Based on Imbalanced Classification for Predicting Outliers in Electricity Demand Time Series, Energies, 9 (2016) 752-752. [http://dx.doi.org/10.3390/en9090752]
[20] A. Capozzoli, F. Lauro, I. Khan, Fault detection analysis using data mining techniques for a cluster of smart office buildings, Expert Systems with Applications, 42 (2015) 4324-4338. [http://dx.doi.org/10.1016/j.eswa.2015.01.010]
[21] A. Ahmad, M. Khan, A. Paul, S. Din, M.M. Rathore, G. Jeon, G.S. Chio, Towards modeling and optimization of features selection in Big Data based social Internet of Things, Future Generation Computer Systems, (2017). [http://dx.doi.org/https://doi.org/10.1016/j.future.2017.09.028]
[22] B. Balaji, J. Xu, A. Nwokafor, R. Gupta, Y. Agarwal, Sentinel: occupancy based HVAC actuation using existing WiFi infrastructure within commercial buildings, in: Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems, ACM, Roma, Italy, 2013, pp. 1-14. [http://dx.doi.org/10.1145/2517351.2517370]
[23] J.H. Reif, Synthesis of parallel algorithms, Morgan Kaufmann Publishers Inc., 1993. [http://dl.acm.org/citation.cfm?id=562546]
[24] A. Ali, G.A. Shah, J. Arshad, Energy efficient techniques for M2M communication: A survey, Journal of Network and Computer Applications, 68 (2016) 42-55. [http://dx.doi.org/10.1016/j.jnca.2016.04.002]
[25] J.-M. Adamo, Data mining for association rules and sequential patterns: sequential and parallel algorithms, Springer Science & Business Media, 2012. [http://books.google.es/books?hl=es&lr=&id=CXPjBwAAQBAJ&oi=fnd&pg=PA49&dq=parallel+and+sequential+implementations+data+mining&ots=X9oC3aiqq0&sig=YqLrFUfswlReoT4p0mucYz-bORU]
[26] L.J. Eshelman, The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination, in: R. Gregory J.E (Ed.) Foundations of Genetic Algorithms, Elsevier, 1991, pp. 265-283. [http://dx.doi.org/10.1016/B978-0-08-050684-5.50020-3]
[27] D. Kanani, K. Shah, R.G. Vaishnav, Cloud Computing–Task Scheduling based on Modified CHC Algorithm, environment, 1 (2016) 2. [http://www.irjet.net/archives/V3/i7/IRJET-V3I791.pdf]
[28] J. Derrac, F. Chiclana, S. García, F. Herrera, Evolutionary fuzzy k-nearest neighbors algorithm using interval-valued fuzzy sets, Information Sciences, 329 (2016) 144-163. [http://dx.doi.org/10.1016/j.ins.2015.09.007]
[29] E. Parras-Gutierrez, V.M. Rivas, J.J. Merelo, A Radial Basis Function Neural Network-Based Coevolutionary Algorithm for Short-Term to Long-Term Time Series Forecasting, in: K. Madani, A. Dourado, A. Rosa, J. Filipe, J. Kacprzyk (Eds.) Computational Intelligence: Revised and Selected Papers of the International Joint Conference, IJCCI 2013, Vilamoura, Portugal, September 20-22, 2013, Springer International Publishing, Cham, 2016, pp. 121-136. [http://dx.doi.org/10.1007/978-3-319-23392-5_7]
[30] V. Paramasivam, T.S. Yee, S.K. Dhillon, A.S. Sidhu, A methodological review of data mining techniques in predictive medicine: An application in hemodynamic prediction for abdominal aortic aneurysm disease, Biocybernetics and Biomedical Engineering, 34 (2014) 139-145. [http://dx.doi.org/10.1016/j.bbe.2014.03.003]
[31] F. Azuaje, Computational models for predicting drug responses in cancer research, Briefings in Bioinformatics, 18 (2017) 820-829. [http://dx.doi.org/10.1093/bib/bbw065]
[32] J. Fan, F. Han, H. Liu, Challenges of Big Data Analysis, National science review, 1 (2014) 293-314. [http://dx.doi.org/10.1093/nsr/nwt032]
[33] H.C. Jung, J.S. Kim, H. Heo, Prediction of building energy consumption using an improved real coded genetic algorithm based least squares support vector machine approach, Energy and Buildings, 90 (2015) 76-84. [http://dx.doi.org/10.1016/j.enbuild.2014.12.029]

[34] C. Deb, F. Zhang, J. Yang, S.E. Lee, K.W. Shah, A review on time series forecasting techniques for building energy consumption, Renewable and Sustainable Energy Reviews, 74 (2017) 902-924. [http://dx.doi.org/10.1016/j.rser.2017.02.085]
[35] X. Qiu, P.N. Suganthan, G.A.J. Amaratunga, Ensemble incremental learning Random Vector Functional Link network for short-term electric load forecasting, Knowledge-Based Systems, 145 (2018) 182-196. [http://dx.doi.org/10.1016/j.knosys.2018.01.015]
[36] X. Qiu, Y. Ren, P.N. Suganthan, G.A.J. Amaratunga, Empirical Mode Decomposition based ensemble deep learning for load demand time series forecasting, Applied Soft Computing, 54 (2017) 246-255. [http://dx.doi.org/10.1016/j.asoc.2017.01.015]
[37] M. Benedetti, V. Cesarotti, V. Introna, J. Serranti, Energy consumption control automation using Artificial Neural Networks and adaptive algorithms: Proposal of a new methodology and case study, Appl Energ, 165 (2016) 60-71. [http://dx.doi.org/10.1016/j.apenergy.2015.12.066]
[38] E. Cadenas, W. Rivera, R. Campos-Amezcua, R. Cadenas, Wind speed forecasting using the NARX model, case: La Mata, Oaxaca, México, Neural Computing and Applications, 27 (2016) 2417-2428. [http://dx.doi.org/10.1007/s00521-015-2012-y]
[39] Z. Xian, C. Ka Wing, Y. Xuesen, Z. Yangyang, Y. Kexin, W. Guibin, A comparison study on electric vehicle growth forecasting based on grey system theory and NAR neural network, in: 2016 IEEE International Conference on Smart Grid Communications (SmartGridComm), 2016, pp. 711-715. [http://dx.doi.org/10.1109/SmartGridComm.2016.7778845]
[40] A. Afram, F. Janabi-Sharifi, A.S. Fung, K. Raahemifar, Artificial neural network (ANN) based model predictive control (MPC) and optimization of HVAC systems: A state of the art review and case study of a residential HVAC system, Energy and Buildings, 141 (2017) 96-113. [http://dx.doi.org/10.1016/j.enbuild.2017.02.012]
[41] J. Wang, J. Wang, W. Fang, H. Niu, Financial time series prediction using elman recurrent random neural networks, Computational intelligence and neuroscience, 2016 (2016). [http://dx.doi.org/10.1155/2016/4742515]
[42] S. Kelo, S. Dudul, A wavelet Elman neural network for short-term electrical load prediction under the influence of temperature, International Journal of Electrical Power & Energy Systems, 43 (2012) 1063-1071. [http://dx.doi.org/10.1016/j.ijepes.2012.06.009]
[43] M. Elkaref, B. Bohnet, A Simple LSTM model for Transition-based Dependency Parsing, arXiv preprint (2017). [http://arxiv.org/pdf/1708.08959]
[44] C. Liang, W. Zhen, W. Gang, Application of LSTM Networks in Short-Term Power Load Forecasting Under the Deep Learning Framework, Electric Power Information and Communication Technology, 15 (2017) 8-11. [http://www.dlxxtx.com/CN/article/downloadArticleFile.do?attachType=PDF&id=1380]
[45] T. Linzen, E. Dupoux, Y. Goldberg, Assessing the ability of LSTMs to learn syntax-sensitive dependencies, arXiv preprint arXiv:1611.01368, (2016). [http://arxiv.org/pdf/1611.01368]
[46] K. Wang, C. Deng, J. Li, Y. Zhang, X. Li, M. Wu, Hybrid methodology for tuberculosis incidence time-series forecasting based on ARIMA and a NAR neural network, Epidemiology & Infection, 145 (2017) 1118-1129. [http://dx.doi.org/10.1017/S0950268816003216]
[47] W. Wu, S. An, J. Guo, P. Guan, Y. Ren, L. Xia, B. Zhou, Application of nonlinear autoregressive neural network in predicting incidence tendency of hemorrhagic fever with renal syndrome, Zhonghua liu xing bing xue za zhi= Zhonghua liuxingbingxue zazhi, 36 (2015) 1394-1396. [http://europepmc.org/abstract/med/26850398]
[48] A. Zavadskaya, Artificial Intelligence in Finance: Forecasting Stock Market Returns Using Artificial Neural Networks (Available on Internet), (2017). [http://helda.helsinki.fi/dhanken/bitstream/handle/123456789/170154/zavadskaya.pdf?sequence=1]
[49] A. Verma, I. Kaur, A. Kaur, Algorithmic approach to data mining and classification techniques, Indian Journal of Science and Technology, 9 (2016). [http://www.indjst.org/index.php/indjst/article/view/88874/71913]

[50] G. Düzenli, RFID card security for public transportation applications based on a novel neural network analysis of cardholder behavior characteristics, Turkish Journal of Electrical Engineering & Computer Sciences, 23 (2015) 1098-1110. [http://journals.tubitak.gov.tr/elektrik/issues/elk-15-23-4/elk-23-4-13-1306-96.pdf]
[51] C. Yang, H. Li, Y. Rezgui, I. Petri, B. Yuce, B. Chen, B. Jayan, High throughput computing based distributed genetic algorithm for building energy consumption optimization, Energy and Buildings, 76 (2014) 92-101. [http://dx.doi.org/10.1016/j.enbuild.2014.02.053]
[52] I. Petri, H. Li, Y. Rezgui, Y. Chunfeng, B. Yuce, B. Jayan, A modular optimisation model for reducing energy consumption in large scale building facilities, Renewable and Sustainable Energy Reviews, 38 (2014) 990-1002. [http://dx.doi.org/10.1016/j.rser.2014.07.044]
[53] M. Ibrahim, S. Jemei, G. Wimmer, D. Hissel, Nonlinear autoregressive neural network in an energy management strategy for battery/ultra-capacitor hybrid electrical vehicles, Electric Power Systems Research, 136 (2016) 262-269. [http://dx.doi.org/10.1016/j.epsr.2016.03.005]
[54] G.I. Nagy, G. Barta, S. Kazi, G. Borbély, G. Simon, GEFCom2014: Probabilistic solar and wind power forecasting using a generalized additive tree ensemble approach, International Journal of Forecasting, 32 (2016) 1087-1093. [http://dx.doi.org/10.1016/j.ijforecast.2015.11.013]
[55] J.L. Elman, Finding Structure in Time, Cognitive Science, 14 (1990) 179-211. [http://dx.doi.org/10.1207/s15516709cog1402_1]
[56] A. Kose, E. Petlenkov, System identification models and using neural networks for Ground Source Heat Pump with Ground Temperature Modeling, in: 2016 International Joint Conference on Neural Networks (IJCNN), 2016, pp. 2850-2855. [http://dx.doi.org/10.1109/IJCNN.2016.7727559]
[57] G. Bao, Q. Lin, D. Gong, H. Shao, Hybrid Short-term Load Forecasting Using Principal Component Analysis and MEA-Elman Network, in: D.-S. Huang, K. Han, A. Hussain (Eds.) Intelligent Computing Methodologies: 12th International Conference, ICIC 2016, Lanzhou, China, August 2-5, 2016, Proceedings, Part III, Springer International Publishing, Cham, 2016, pp. 671-683. [http://dx.doi.org/10.1007/978-3-319-42297-8_62]
[58] S. Qin, J. Wang, J. Wu, G. Zhao, A hybrid model based on smooth transition periodic autoregressive and Elman artificial neural network for wind speed forecasting of the Hebei region in China, International Journal of Green Energy, 13 (2016) 595-607. [http://dx.doi.org/10.1080/15435075.2014.961462]
[59] C.-T. Chu, C.-H. Chang, T.-J. Chang, J.-X. Liao, Elman neural network identify elders fall signal base on second-order train method, in: Next Generation Electronics (ISNE), 2017 6th International Symposium on, IEEE, 2017, pp. 1-4. [http://dx.doi.org/10.1109/ISNE.2017.7968722]
[60] P. Li, Y. Li, Q. Xiong, Y. Chai, Y. Zhang, Application of a hybrid quantized Elman neural network in short-term load forecasting, International Journal of Electrical Power & Energy Systems, 55 (2014) 749-759. [http://dx.doi.org/10.1016/j.ijepes.2013.10.020]
[61] S. Ekici, S. Yildirim, M. Poyraz, A transmission line fault locator based on Elman recurrent networks, Applied Soft Computing, 9 (2009) 341-347. [http://dx.doi.org/10.1016/j.asoc.2008.04.011]
[62] M. Cuéllar, M. Delgado, M. Pegalajar, Multiobjective evolutionary optimization for Elman recurrent neural networks, applied to time series prediction, Fuzzy Economic Review, 10 (2005) 17-33. [http://search.proquest.com/docview/228956321]
[63] M. Delgado, M.C. Pegalajar, M.P. Cuéllar, Memetic evolutionary training for recurrent neural networks: An application to time-series prediction, Expert Systems, 23 (2006) 99-114. [http://dx.doi.org/10.1111/j.1468-0394.2006.00327.x]
[64] V. Singh, K. Tiwari, Prediction of GreenHouse Micro-Climate using Artificial Neural Network, Applied Ecology and Environmental Research, 15 (2017) 767-778. [http://www.aloki.hu/pdf/1501_767778.pdf]
[65] K. Greff, R.K. Srivastava, J. Koutník, B.R. Steunebrink, J. Schmidhuber, LSTM: A Search Space Odyssey, IEEE Transactions on Neural Networks and Learning Systems, 28 (2017) 2222-2232. [http://dx.doi.org/10.1109/TNNLS.2016.2582924]
[66] F.J. Ordóñez, D. Roggen, Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition, Sensors, 16 (2016) 115. [http://dx.doi.org/10.3390/s16010115]

[67] X. Lu, Y. Chen, X. Li, Hierarchical Recurrent Neural Hashing for Image Retrieval With Hierarchical Convolutional Features, IEEE Transactions on Image Processing, 27 (2018) 106-120. [http://dx.doi.org/10.1109/TIP.2017.2755766]
[68] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation, 9 (1997) 1735-1780. [http://www.mitpressjournals.org/doi/pdfplus/10.1162/neco.1997.9.8.1735]
[69] D.L. Marino, K. Amarasinghe, M. Manic, Building energy load forecasting using Deep Neural Networks, in: IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society, 2016, pp. 7046-7051. [http://dx.doi.org/10.1109/IECON.2016.7793413]
[70] K. He, J. Sun, Convolutional neural networks at constrained time cost, in: Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on, IEEE, 2015, pp. 5353-5360. [http://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/He_Convolutional_Neural_Networks_2015_CVPR_paper.pdf]
[71] H.G. Lee, C.Y. Yi, D.E. Lee, D. Arditi, An Advanced Stochastic Time- Cost Tradeoff Analysis Based on a CPM- Guided Genetic Algorithm, Computer- Aided Civil and Infrastructure Engineering, 30 (2015) 824-842. [http://dx.doi.org/10.1111/mice.12148]
[72] T. Blickle, L. Thiele, A comparison of selection schemes used in genetic algorithms, (1995). [http://pdfs.semanticscholar.org/fef8/1135f587851f19fe515cb8eb3812e3706b27.pdf]
[73] F. Herrera, M. Lozano, J.L. Verdegay, Tackling Real-Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis, Artificial Intelligence Review, 12 (1998) 265-319. [http://dx.doi.org/10.1023/a:1006504901164]
[74] H. Junninen, H. Niska, K. Tuppurainen, J. Ruuskanen, M. Kolehmainen, Methods for imputation of missing values in air quality data sets, Atmospheric Environment, 38 (2004) 2895-2907. [http://dx.doi.org/10.1016/j.atmosenv.2004.02.026]