INTELLIGENT ELECTRICAL MULTI OUTLETS CONTROLLED AND ACTIVATED BY A DATA MINING ENGINE ORIENTED TO BUILDING ELECTRICAL MANAGEMENT

Alessandro Massaro, Giacomo Meuli, Angelo Galiano

Dyrecta Lab, IT Research Laboratory, Via Vescovo Simplicio, 45, 70014 Conversano (BA), Italy

ABSTRACT

In the proposed paper are discussed results of an industry project concerning energy management in building. Specifically the work analyses the improvement of electrical outlets controlled and activated by a logic unit and a data mining engine. The engine executes a Long Short-Terms Memory (LSTM) neural network algorithm able to control, to activate and to disable electrical loads connected to multiple outlets placed into a building and having defined priorities. The priority rules are grouped into two level: the first level is related to the outlet, the second one concerns the loads connected to a single outlet. This algorithm, together with the prediction processing of the logic unit connected to all the outlets, is suitable for alerting management for cases of threshold overcoming. In this direction is proposed a flow chart applied on three for three outlets and able to control load matching with defined thresholds. The goal of the paper is to provide the reading keys of the data mining outputs useful for the energy management and diagnostic of the electrical network in a building. Finally in the paper are analyzed the correlation between global active power, global reactive power and energy absorption of loads of the three intelligent outlet. The prediction and the correlation analyses provide information about load balancing, possible electrical faults and energy cost optimization.

KEYWORDS

Intelligent Electrical Outlets, Energy Management, Load Monitoring and Piloting, Smart Grids, Energy Routing, Data Mining, KNIME, Long Short-Term Memory (LSTM), Neural Network, Correlation Matrix.

1.INTRODUCTION: STATE OF THE ART AND MAIN PROJECT SPECIFICATIONS

In [1], some authors highlighted the importance of Machine Learning (ML) algorithms for the planning of the activation of electric utilities starting from the distribution of electrical energy and by analyzing predictive data [1]. Other researchers studied accurately the topic of electric energy management in the smart home, by analyzing the combined architecture of the electric power transmission together with the communication network using smart meters [2]. The use of the smart meters is useful for the real time reading of the electrical power providing, though an external control unit, information about the "status" of the power consumption and reconstruction of periodic load curve. Other scientific studies focused the attention on peak load prediction [3], and on price prediction [4]. A complete home energy management model has been presented in [5]: in this model are defined the optimal consumption strategies, planning at the same time load control activities. Important topics for the energy consumption optimization strategies are the

DOI :10.5121/ijscai.2018.7401

daily and annual energy losses simulations [6]: in this direction important inputs for the calculations are load profiles (load curves), the number of daily use of electric loads (number of users / day), the probability of utilization, and the percentage of loads per home. Important tools suitable for residential building energy management system are data mining tools adopted in literature for non-intrusive applications [7], for the time forecasting prediction of the electricity consumption [8], and for the loads control [9]. It is important to observe that for domestic applications the total consumer load is defined as a combination of small loads to be controlled by piloted switches [10], where groups of small loads can be controlled by a multi-outlet device. Load management must be performed using a load controller [11], consisting of a PC-programmable microcontroller. Another applications involving multi-outlets are the management of electricity produced by photovoltaic panels [12], and energy management improved by sensors, MySQL databases and graphical interfaces [13]. The concept of intelligent electrical outlets has been developed in [14], in which a system server center administrates and controls by web services the electrical loads. Intelligent systems combined with sensors and smart meters can be integrated into more complex systems such as those reported in [15]. Smart sockets (electrical outlets) can also be applied to multi-floor buildings by configuring wireless nodes for control [16]. Energy building management by data mining improving decision intelligence has been analyzed in [17]. In particular data mining clustering approach has been adopted for energy usage profiles of domestic residences [18]. The grouping of electrical users by data mining dendograms has been applied to associate electrical utilities used during the week [19]. Other researchers analyzed how intelligent electrical outlets can also be used to prevent fires (overheating alerting) or electrical malfunctions [20]. In [21] Artificial Neural Networks (ANNs) have been adopted to predict energy consumption, besides in [22] the Long-Short-Term-Memory (LSTM) based Recurrent Neural Network (RNN) provided a good accurate method for complex electric load time series forecasting also for long forecasting horizon. Concerning the electronic circuits able to process and to transmit data, some researchers studied the possibility to implement Arduino [23] or Raspberry [24] technologies, thus suggesting to improve a logic unit of electrical outlets by microcontrollers. Following the state of the art has been formulated the idea of the project sketched in the architecture of Fig. 1decribed below:

- controlled electrical outlets (Outlet 1, Outlet 2, etc.) transmit data in cloud by a personal computer (PC) having an USB interface, an intelligent unit able to detect electrical current and electrical power ($P(t)=V(t)\cdot I(t)$) and to control the load activations, and an ethernet interface connecting outlet to internet or intranet network;
- a data mining engine implementing data mining algorithms.

The application of the data mining algorithms will therefore assist to:

- optimize the activation and the deactivation of electrical loads basing on consumption prediction (alert thresholds are enabled by the data mining engine which will deactivate loads in threshold overcoming cases);
- optimize costs and reduce energy waste, taking into account energy tariff conditions;
- define dynamically the thresholds by considering tariffs and predicted consumptions;
- predict electrical faults by an advanced analysis of the anomalies found in the network connected to the multi-outlet system.

The paper is structured as follows:

- design principles of the LSTM neural network prototype systems implemented to optimize an electrical building network, describing logics to activate and to deactivate loads connected to different electrical outlets;

- the testing of the LSTM algorithm by considering a dataset of measured values of a network of three electrical outlets providing measured data;
- a complete scenario to understand prediction of global active power of the building;
- a correlation analysis between different attributes including active and reactive power, explaining the status of the analyzed building electrical network.

The highlights of the proposed paper are the following ones:

1- application of the load prediction theory discussed in [25], together with LSTM electrical power prediction in order to increase the reliability of the prediction;

2- definition of the flowchart able to enable and disable loads characterized by two priority level (priority of the electrical outlet and load priority);

3- testing of the LSTM model showing good performance;

4- explanation of the correlation of reactive power with active one by identifying inefficient outlets.

Below is summarized the features and the requirements of the whole proposed mode :

Features and requirements	Description
Electrical load prediction (prediction 1)	Slope behavior prediction of the load (processing of the intelligent unit)
LSTM model (prediction 2)	Training and testing of the LSTM model based on real dataset found online
Prototype model	Testing based on 3 electrical outlets having different priority
Implementation of the priority roles and enabling /disabling procedures	Flowchart describing two priority levels of electrical loads and criteria to deactivate/activate loads by processing prediction results (prediction 1 and prediction 2)
Post processing	Analysis of correlations between all the attribute analyzed in the model in order to check possible load malfunctions

Table 1. Table of the work requirements and features.

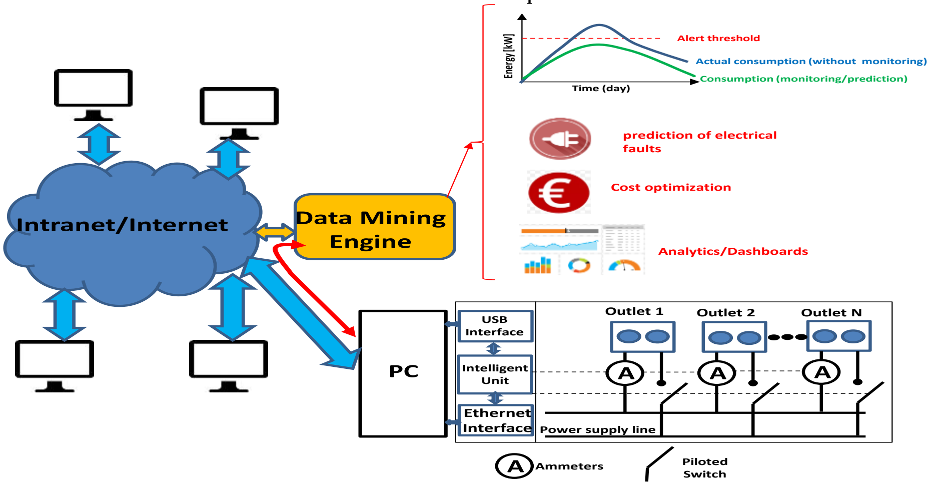


Figure 1. Schematic system architecture of intelligent electrical outlets integrated into internet/intranet network.

2. Logics and Design of the data mining engine managing intelligent electrical outlets

In this section are discussed the logics of electrical load management and the design procedures of the data mining engine reported in Fig. 1.

2.1 DESIGN OF THE PROTOTYPE SYSTEM

The main design concept is illustrated in Fig. 2 where are identified three main design stages such as:

1. Measurement. A time dependent total power is measured by the total current detection;

2. Evaluation. An algorithm, enabled by the intelligent unit, assesses when the energy consumption is too high, when it is within normal limits or when it is significantly lower. Depending on the following three cases, it checks:

- when the energy consumed is at the limit (the algorithm intervenes on the load management by deactivating the outlet to which the least priority device is connected);
- when the energy consumed is too high (the algorithm intervenes by interrupting the current of outlets having devices with lower priority by restoring the maximum permissible power range);
- when the energy consumed is significantly lower (the algorithm intervenes by rehabilitating the outlets previously disabled having higher priority);

3. management of electrical outlets. According with the evaluation, the system decides which outlets to activate or deactivate, according to the following rules:

-priority: (the outlet which has a lower priority value will typically be the one of minor importance for the application; if an electrical outlet is to be reactivated, the algorithm reorders the priority values in descending order and follows the order of the list);

-absorption entity (indicates the energy absorbed by the single outlet calculated through the evaluation of the current in the last minute);

-availability: indicates the on/off status of the outlet.

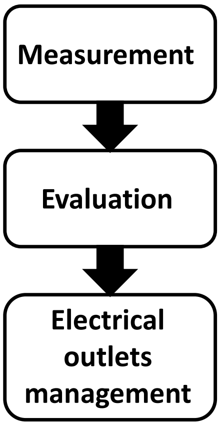


Figure 2. Design concept of the intelligent electrical outlet system.

In Fig. 3 (a) is indicated a typical current trend versus minutes: the main value to process is the total current of the electrical outlet providing the global active power of the whole electrical network. This last parameter is utilised in order to check the overcoming of the threshold limit. In Fig. 3 (b) is plotted an example related to an alerting condition due to the threshold overcoming. The intelligent logic unit executes the algorithm which evaluates the slope of the curve at each minute by comparing it with the slope of the line representing the maximum absorption (see Fig. 4) [25]. The blue line indicates the measured electrical power of an electrical outlet connected with different loads, and the red line indicates the threshold line having a defined slope according with energy management risk. Each power variation during the time is characterized by a slope. If the intelligent unit predicts that the total power overcomes the threshold line, and if also the LSTM network provides the same prediction, will be deactivated no priority load or groups of no priority loads. The slope prediction is estimated by assuming the same slope behaviour of the last minute (measurement of the last minute). By considering a system composed by different electrical outlet of a building it is necessary to predict also the behaviour of the whole energy managed by all the electrical outlets. In this case the concept of priority is extended also to outlet priority by introducing a second level of priority: the first priority level is related to the outlet and the second one concerns the loads of a single outlets. A good design takes into account the following considerations:

- the priority loads are grouped into outlets having the same or similar priority;
- the amperometers and the power meters are placed not only for the measurements of the absorbed energy of an outlet but also for the measurements of main loads of the same outlet;
- the typical load curves of the building in different periods of the years (important for the initial load distribution evaluation).

The predictions of the energy absorbed of each outlet at the time step *t*-1 (indicated in Fig. 5 by the attribute sub-metering) are the input of the Long Short-Term neural network predicting the global active power at the time *t*. The global active power represent the main attribute of the electrical building energy which will be considered for electricity management.

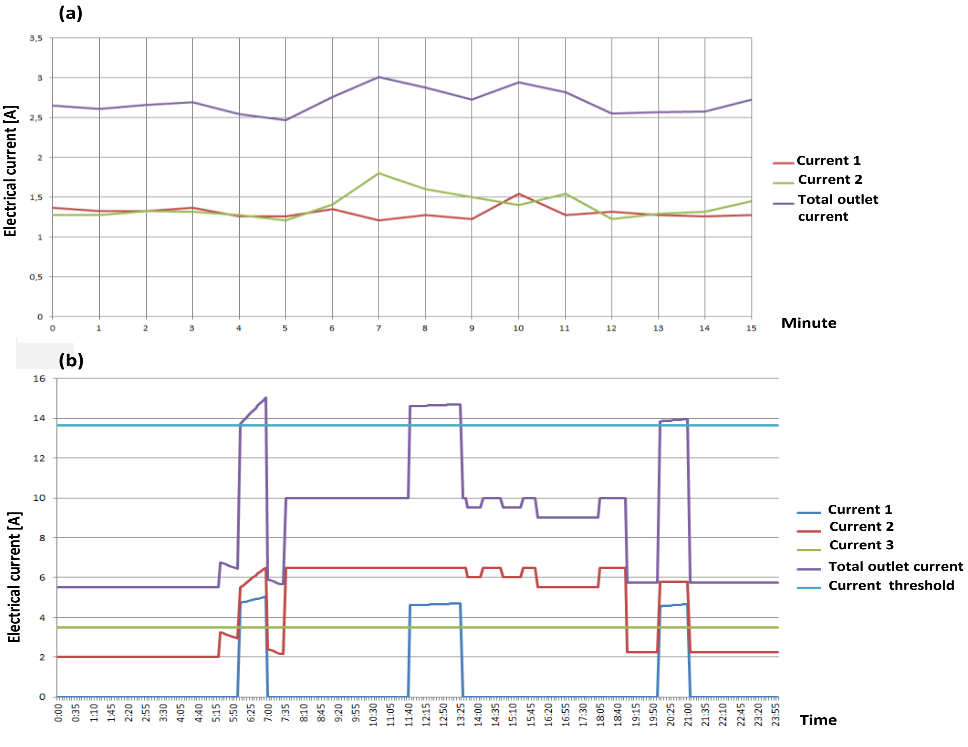


Figure 3. Example of current measurement and evaluation processes of an electrical outlet: (a) outlet 1; (b) outlet 2.

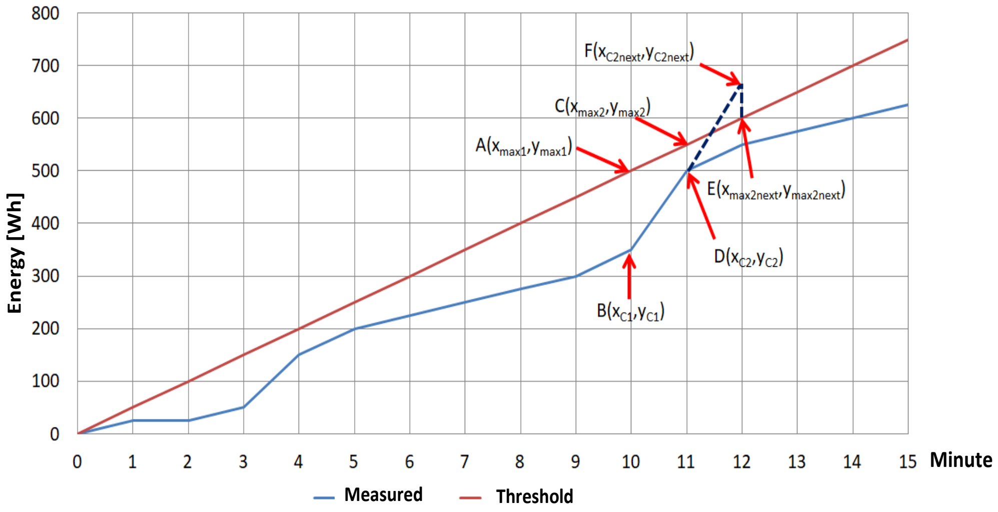


Figure 4. Evaluation of load slope and comparison with the threshold [25]: global energy threshold.

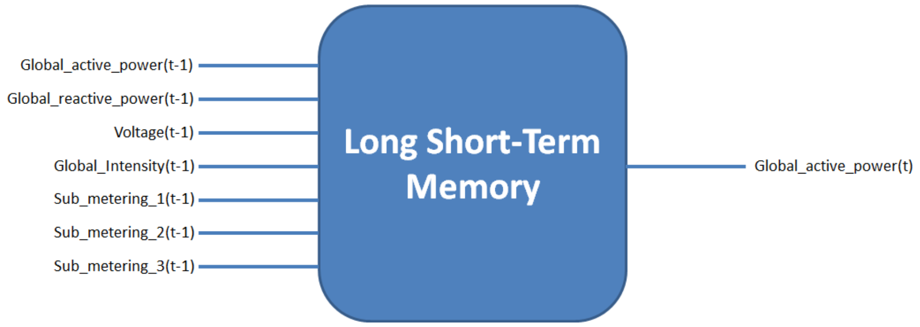


Figure 5. Model of the LSTM predicting global power.

In Fig. 6 and Fig. 7 are illustrated the flowcharts describing the control logic assuming the load priority configuration of table 2 where are reported the priority levels of three electrical outlets and 7 loads. The alerting condition management procedure is explained by the flow chart of Fig. 6: the logic unit predict the slope of the global energy, if the prediction generates an alerting condition will be analysed the prediction of the LSTM algorithm, if also the LSTM output provides a further alerting condition will be activated the logic of load disabling shown in the flow chart of Fig. 7. According to the priority level of table 1 will be disabled before the loads of the outlet 1 having lower priority and successively the loads of the other outlets until the estimated value does not exceed the threshold. This procedure is valid for a precise time *t*. For each time steps will be repeated the disabling procedure. When will be predicted no risk condition will be enable gradually the loads initially disabled by the automatic procedure. If only the prediction of slope provides an alerting condition because the LSTM indicates no risk, for a better security, will be disabled only the first one or two loads having lower priority. the LSTM network is “reinforcing”, and is used to estimate with greater accuracy that there is a certain probability of exceeding the threshold. The proposed procedure can be applied for different time units (minutes, days, weeks, etc.).

Outlet Priority level	**Outlet**	**Load**	
3	1	**Load priority**	**Load**
		7	1
		6	2
2	2	**Load priority**	**Load**
		5	3
		4	4
		3	5
1	3	**Load priority**	**Load**
		2	6
		1	7

Table 2. Example of priority configuration for three electrical outlets and seven loads.

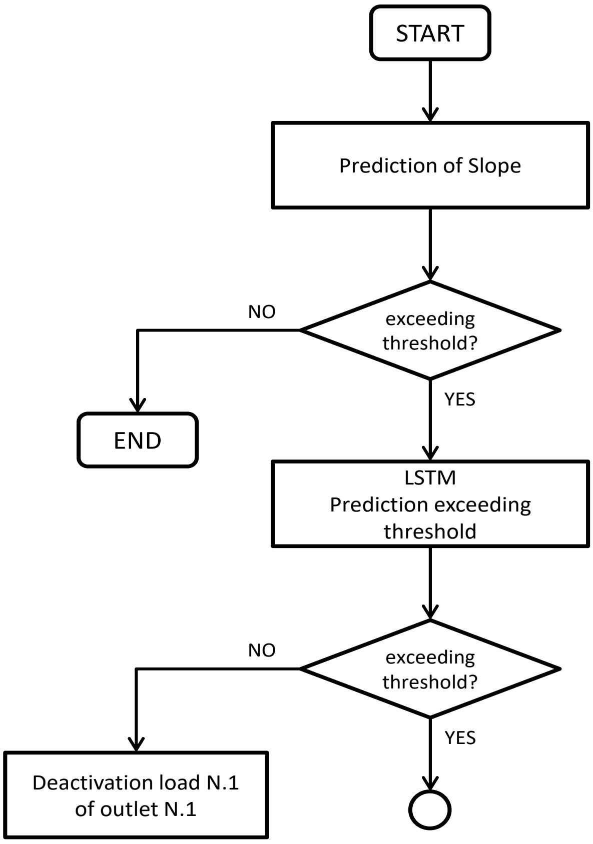


Figure 6. Sequential model for threshold alerting condition.

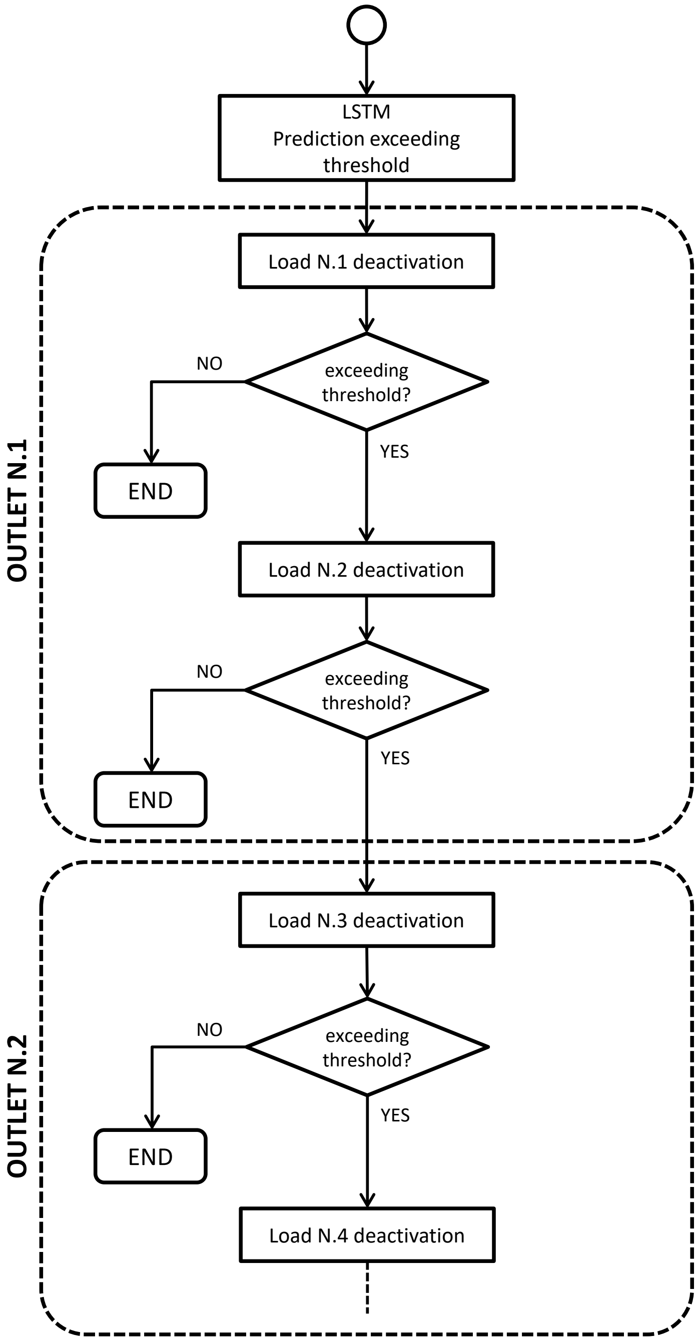


Figure 7. Flowchart of the load disabling procedure.

2.2 EXPERIMENTAL DATASET

In order to check the implemented LSTM algorithm has been adopted the dataset proposed in [26] and [27]. This dataset contains 2075259 measurements related to period between December 2006 and November 2010 (47 months). We provide below more information about the dataset attributes.

- "Global_active_power": indicates the global active power absorbed by the family unit (household global minute-averaged active power), expressed in kW (kilowatt);

- "Global_reactive_power": indicates the global reactive power absorbed by the family unit (household global minute-averaged reactive power), expressed in kW (kilowatt);
- "Voltage": indicates the minute-averaged voltage, expressed in V (volts);
- "Global_intensity": indicates the global current absorbed by the family unit (household global minute-averaged current intensity), expressed in A (ampere);
- "Sub_metering_1": corresponds to the energy absorption of the loads that usually are in "kitchen", therefore they are mainly considered a dishwasher, an oven and a microwave oven, and is expressed in Wh (watt hour of active energy);
- "Sub_metering_2": corresponds to the energy absorption of the loads that usually are in a "laundry" and are mainly part of a washing machine, a dryer, a refrigerator and a light, and is expressed in Wh (watt hour of active energy);
- "Sub_metering_3": corresponds to the energy absorption of an electric water heater and an air conditioner; it is expressed in Wh (watt hour of active energy);
- The formula global_active_power*1000/60- sub_metering_1 -sub_metering_2 -sub_metering_3 represents the active energy consumed every minute (Wh) in the household by electrical equipment not measured in sub-meterings 1, 2 and 3;
- The dataset contains some missing values in the measurements (nearly 1,25% of the rows);
- date: Date in format dd/mm/yyyy;
- time: time in format hh:mm:ss.

The experimental dataset is plotted in Fig. 8 where all the attributes will train the LSTM model of Fig.5. In order to analyse better the dispersion of the measurement for a check of the measurements reliability, the scattering multiple graphical library of the Rapid Miner tool has been adopted. In Fig. 9 are plotted all the dataset attributes by observing a good voltage trend, a typical global power distribution and a clear differentiation of load distributions (the outlet associated to sub_metering_3 it consumes on average more energy than the other outlets). This first check is important in order to verify if the if the priority rule is respected, and if there are some unbalanced cases or malfunctions.

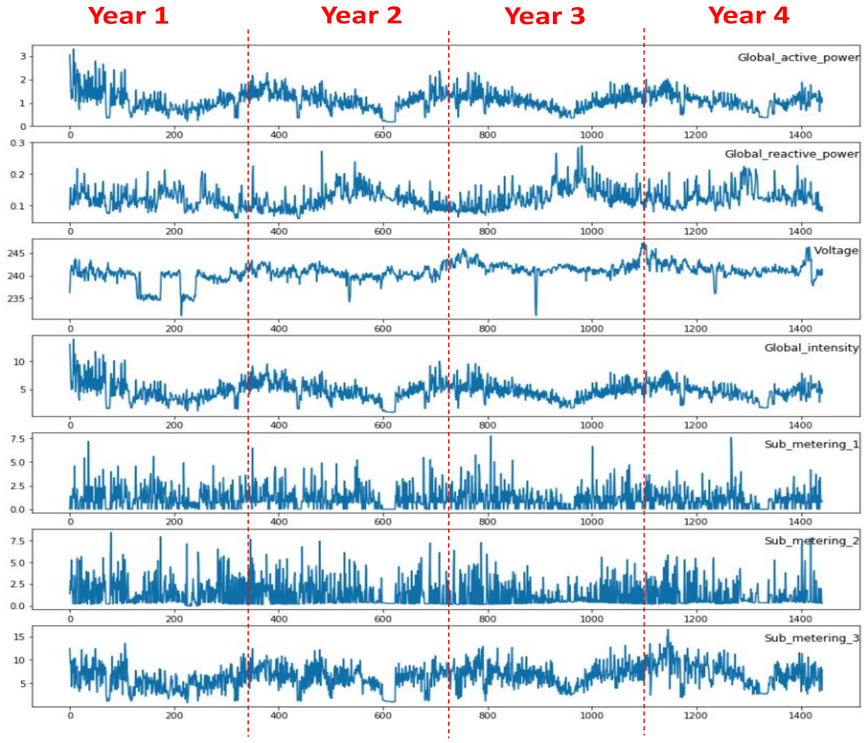


Figure 8. Experimental dataset (daily samples).

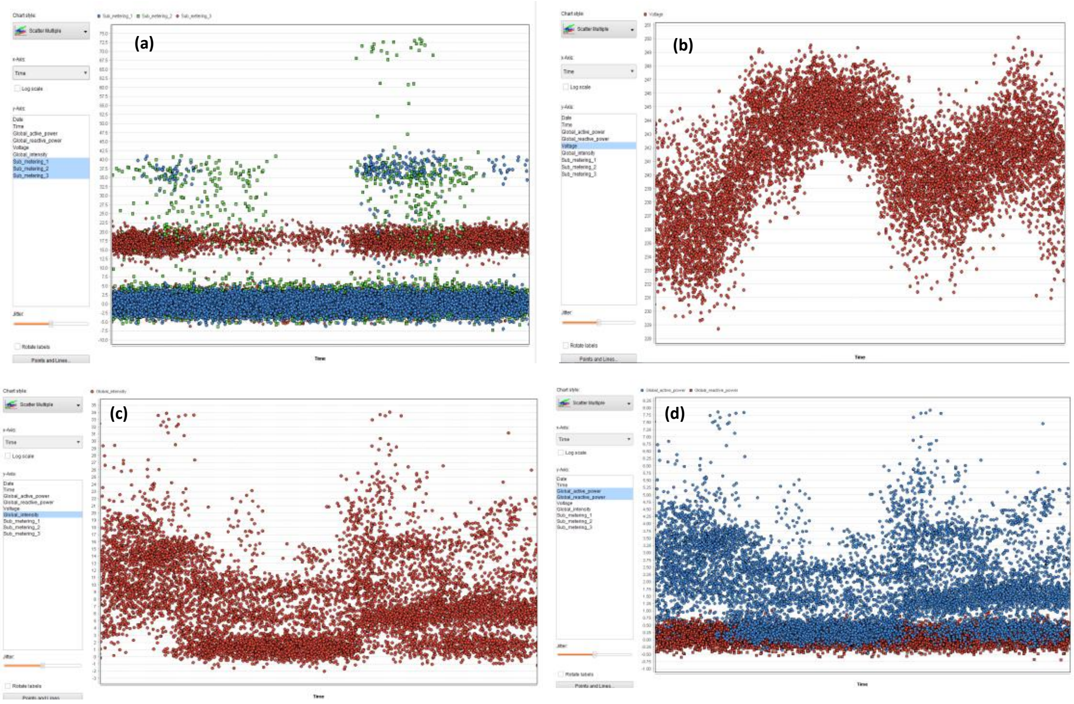


Figure 9. Experimental dataset: scattering multiple library of the Rapid Miner tool.

3. LSTM Implementation and Results

In order to implement the LSTM time series forecasting algorithm it is advisable to choose an appropriate programming language. A suitable language is "python" which offers a series of libraries optimized for Machine Learning and Data Mining. The main libraries used for the LSTM implementation are:

• "Numpy": it is a library that adds support for the management of vectors and for large multidimensional arrays with high level mathematical functions;
• "Matplotlib": it is a library for creating graphs;
• "Seaborn": it is a graphical display library based on matplotlib. It presents a high-level interface for drawing statistical graphs;
• "Pandas": it is a library that provides structures and tools for data analysis. The heart of the library are DataFrame objects, ie 2D data structures indexed both on the columns and on the rows. An object of the DataFrame class can be seen as a SQL table;
• "Scipy": it is an open-source library of mathematical algorithms and tools. It contains modules for optimization, linear algebra, integration, special functions, Fast Fourier Transform, signal and image processing and other tools for science and engineering purposes;
• "Scikit-learn": it is an open-source library for machine learning. It contains classification algorithms, regression, clustering, Support Vector Machine, logistic regression, Bayesian classifier, k-mean and DBSCAN;
• "Keras": it is a high level API for neural networks and for Deep-Learning. It was developed with the aim of allowing a rapid experimentation, ie, being able to move from idea to result in the shortest possible time. Therefore the use of "Keras" allows an easy and fast prototyping, supports convolutional networks and recurring networks. It also has additional features for parallelizing processes using GPUs (strictly CUDA).

Below is reported the python script enabling the importing of the libraries:

```
import sys
import numpy as np
from scipy.stats import randint
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.cross_validation import KFold
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler
from sklearn.pipeline import Pipeline
from sklearn.model_selection import cross_val_score
from sklearn.feature_selection import SelectFromModel
from sklearn import metrics
from sklearn.metrics import mean_squared_error,r2_score

import keras
from keras.layers import Dense
from keras.models import Sequential
from keras.utils import to_categorical
from keras.optimizers import SGD
from keras.callbacks import EarlyStopping
from keras.utils import np_utils

import itertools
from keras.layers import LSTM
from keras.layers.convolutional import Conv1D
from keras.layers.convolutional import MaxPooling1D
from keras.layers import Dropout
```

the dataset is read by the following script:

```
df = pd.read_csv('../input/household_power_consumption.txt', sep=';', parse_dates={'dt' : ['Date', 'Time']},

infer_datetime_format=True, low_memory=False, na_values=['nan','?'], index_col='dt')
```

This operation allows the reading of the dataset and performs a parsing on the "Date" and "Time" columns: in particular it is set as "datetime" format to allow easier management of the dataset tuples representing them according to a "time" -series". Another operation carried out is the conversion of the "missing value" represented as "nan" or "?" And their conversion into "nan" of the numpy type. However, in order to perform a correct LSTM execution it is important to attribute a value to them: generally, the "missing value" is generally placed at their corresponding average value. The following code carries out this assumption and verifies that there are no more "nan" values.

```
droping_list_all=[]

 for j in range(0,7):

 if not df.iloc[:, j].notnull().all():
```

```
droping_list_all.append(j)

droping_list_all
```

The next code performs the replacement of the value and verifies the absence of "nan":

```
for j in range(0,7):

 df.iloc[:,j]=df.iloc[:,j].fillna(df.iloc[:,j].mean())

df.isnull().sum()
```

The Kaggle platform [28] has been adopted for the execution of the LSTM algorithm.

3.1 LSTM Neural Network Predictive Results

Before to execute the LSTM algorithm all attributes have been normalized by the following python script:

```
scaler = MinMaxScaler(feature_range=(0, 1))
scaled = scaler.fit_transform(values)
```

Once the features are selected, the normalized dataset is divided into the following two groups:

- "Train dataset" corresponding to the set of patterns used to train the system;
- "Test dataset" corresponding to the set of patterns adopted to evaluate the final performance of the system.

We chose to train the model on three years of data, and we will test it on the remaining year. The following code "splits" the normalized data in the two groups of "train" and "test":

```
# split dati in train e test
values = reframed.values
n_train_time = 365*24*3
train = values[:n_train_time, :]
test = values[n_train_time:, :]
# the train and test data are divided into algorithm inputs and outputs
train_X, train_y = train[:, :-1], train[:, -1]
test_X, test_y = test[:, :-1], test[:, -1]
# the input reshape is done [samples, timestep, features]
train_X = train_X.reshape((train_X.shape[0], 1, train_X.shape[1]))
test_X = test_X.reshape((test_X.shape[0], 1, test_X.shape[1]))
print(train_X.shape, train_y.shape, test_X.shape, test_y.shape)
```

The LSTM network is characterized as follows:

- 100 neurons number;
- 20% dropout (a random cut of 20% of the input values is made to prevent overfitting);
- the model has been trained according to 20 training periods with a dataset size of 60 samples
- for the evaluation of the error is performed by the "Mean Squarred Error" and the gradient with the "Adam" optimizer

Below is listed the code executing LSTM algorithm:

```
model = Sequential()
model.add(LSTM(100, input_shape=(train_X.shape[1], train_X.shape[2])))
model.add(Dropout(0.2))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam')
# fit network
history = model.fit(train_X, train_y, epochs=20, batch_size=60, validation_data=(test_X, test_y),
verbose=2, shuffle=False)

# model training

model = Sequential()
model.add(LSTM(100, input_shape=(train_X.shape[1], train_X.shape[2])))
model.add(Dropout(0.2))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam')
# addestramento rete
history = model.fit(train_X, train_y, epochs=20, batch_size=50, validation_data=(test_X, test_y),
verbose=2, shuffle=False)

# prediction
yhat = model.predict(test_X)
test_X = test_X.reshape((test_X.shape[0], 7))

# inversion scale for the forecast
inv_yhat = np.concatenate((yhat, test_X[:, -6:]), axis=1)
inv_yhat = scaler.inverse_transform(inv_yhat)
inv_yhat = inv_yhat[:,0]
# inversion scale current values
test_y = test_y.reshape((len(test_y), 1))
inv_y = np.concatenate((test_y, test_X[:, -6:]), axis=1)
inv_y = scaler.inverse_transform(inv_y)
inv_y = inv_y[:,0]
# Root Mean Square Error evaluation
rmse = np.sqrt(mean_squared_error(inv_y, inv_yhat))
plt.plot(inv_y, label="True Value")
plt.plot(inv_yhat, label="Predicted Value")
plt.legend(['Valore vero', Predicted Value], loc='upper right')
plt.show()
print('Test RMSE: %.3f' % rmse)
```

In Fig. 10 (a), Fig. 10 (b), Fig. 10 (c), and Fig. 10 (d) are illustrated the output results concerning the comparison between actual measured value and predicted ones of global active power, for 100, 200, 400 and 1000 samples, respectively. A good matching between actual and predicted results is observed. In Fig.10 is explained in details how predicted results can be analysed: by focusing the attention on the first 10 samples and by assuming that are available the first five measurements, the predicted value will refer to the next predicted day (sample 6 of Fig. 11). By waiting the measurement of the sixth day will be possible to estimate the prediction error as the difference between measured value and predicted one. The plot of Fig. 11 is fundamental in order to check the reliability of the predicted slope behaviour of the global energy. LSTM could be adopted also for medium and long period prediction.

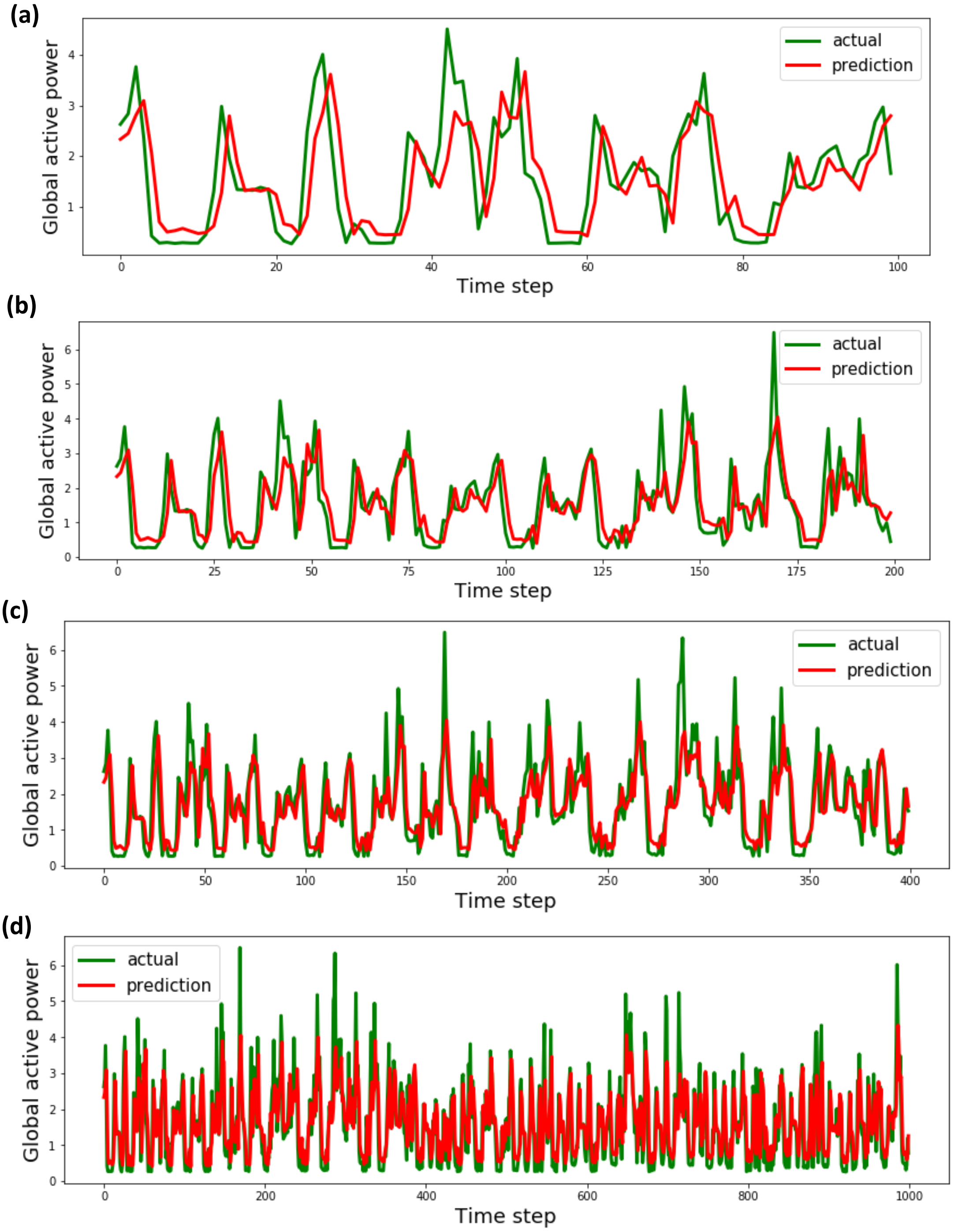


Figure 10. Outputs of the LSTM algorithm: comparison between measured value (actual) and predicted ones plotted for different number of samples.

The error estimation has been performed by the model loss of Fig. 12 representing loss (the training loss is the average of the losses over each batch of training data) versus epoch (an epoch is an iteration on all the data of train; one epoch is when the entire dataset is passed forward and backward through the neural network only once): from the plot of loss, we can see that the model has comparable performance on both train and validation datasets (labeled test). In Fig. 12 a good fit is observed, where the train and validation loss decrease and stabilize around the same value after few epochs (the performance of the model is good on both the train and validation sets). The model loss evaluation has been performed in 21 seconds (train on 8760 samples, validated on 25828 samples for 20 epochs).

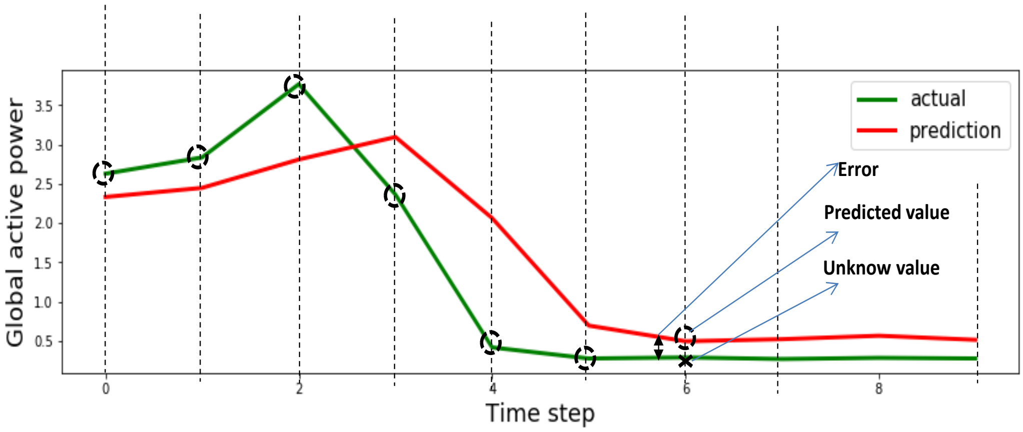


Figure 11. Explanation of the predicted global active power and estimation of the prediction error (one sample refers to a day).

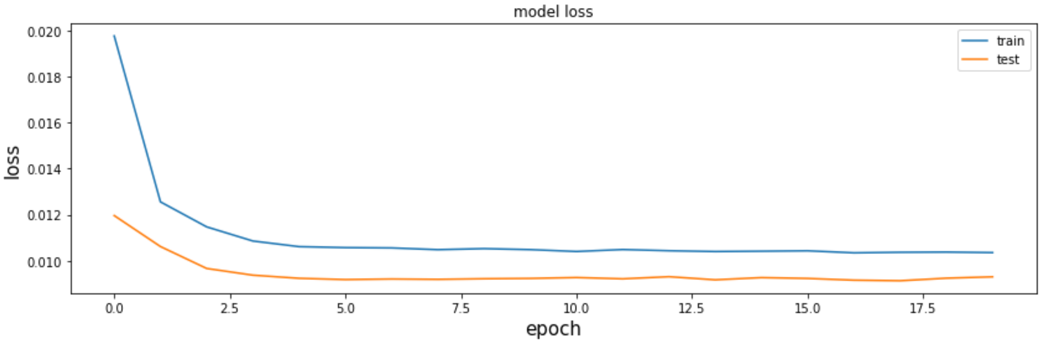


Figure 12. Model loss.

The recurrent neural network LSTM is then indicated for the prediction of electric loads: using a suitable dataset with high samples it is possible to obtain time forecasts and trends that matching with real surveys. The use of this algorithm allows to optimize and manage various scenarios such as:

Activation and deactivation of electrical loads following the predictive assessment of energy consumption;

Reduction of costs and energy wastage according to the fundamental time slots if a multi-hour rate is usually divided as follows:

- ○ peak band: this is the band with the highest price, from Monday to Friday from 8:00 to 19:00 and corresponds to the moment of maximum energy demand;
- ○ Mid-range: prices below the peak range and goes from Monday to Friday, from 7:00 to 8:00 and from 19:00 to 23:00, and Saturday, from 7:00 to 23:00: 00;
- ○ Off-peak: it's the lowest price range and goes from Monday to Saturday, from midnight to 7.00am and from 11.00pm to 12.00pm, including all Sundays and public holidays. The algorithm is organized on the basis of timestamp so the prediction can be made also based on the range of hours and days, corresponding with the time slots, instead of the evaluation based on days, months, quarters and years. All this can be done using the library function for managing the date pandas "pandas" date_range.

Prediction of hypothetical electrical faults by evaluating the excessive absorptions by comparing them with the maximum specifications of the system (for example the system can have a maximum power of 3000W and a maximum current ~ 13.63A). Therefore considering the selected dataset, particular attention will be paid to the features "Global_active_power" ie to the active power of the system and "Global_intensity" which indicates the total current absorbed.
Another strategy is to adopt LSTM by focusing the attention on defined periods: for example if it is important to predict the energy in the next Monday, you could train the model by only the energy measurement of all the past Monday. Other scenarios and data interpretation are obtained by the correlation analysis of attributes.

3.2 Correlation Matrix Results and Final Comments about Work Results

By correlation we mean a relationship between two statistical variables such that each value of the first variable corresponds with a certain regularity a value of the second, that is the tendency of a variable to vary according to another. The degree of correlation between two variables is expressed through the so-called "correlation indices". These indices take values between -1 and 1:

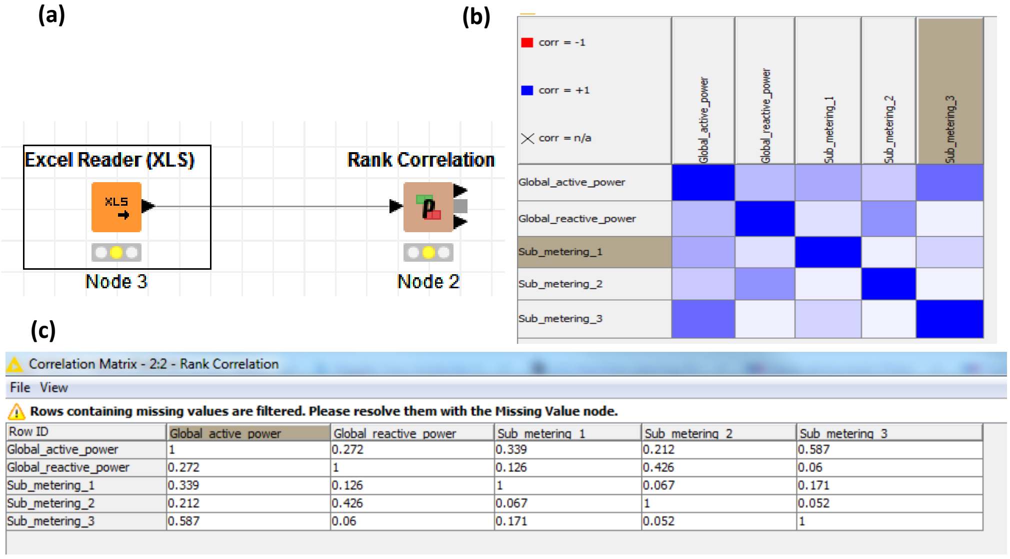
- "-1": the variables are considered inversely correlated;
- "1": the variables are considered absolutely correlated ie when the variation of a variable corresponds to a variation strictly dependent on the other;
- "0": indicates a lack of correlation.

The correlation coefficients are derived from the correlation indexes taking into account the magnitudes of the deviations from the mean. In particular, the Pearson-Bravais correlation coefficient is calculated as the ratio between the covariance of the two variables and the product of their standard deviations:

$$-1 \leq \rho_{xy} = \frac{\sigma_{xy}}{\sigma_x \sigma_y} = \frac{\sum_{i=1}^{n}(x_i - \mu_x)(y_i - \mu_y)}{\sqrt{\sum_{i=1}^{n}(x_i - \mu_x)^2}\sqrt{\sum_{i=1}^{n}(y_i - \mu_y)^2}} \leq +1 \tag{1}$$

The correlation relationships between the main attribute of the experimental dataset have been estimated by the Konstanz Information Miner (KNIME) “Rank Correlation” algorithm. Calculates for each pair of selected columns a correlation coefficient, i.e. a measure of the correlation of the two variables. Below is described the adopted algorithm. All measures are based on the rank of the cells. The rank of a cell value refers to its position in a sorted list of all entries. All correlation can be calculated on the attributes load as DataColumn in the local repository. The algorithm estimates the Spearman's rank correlation coefficient which is a statistical measure of the strength of a monotonic relationship between paired data. The Spearman correlation coefficient is defined as the Pearson correlation coefficient between the ranked variables [29], where the monotonic relationship is characterised by a relationship between ordered sets that preserves the given order, i.e., either never increases or never decreases as its independent variable increases. The value of this measure ranges from -1 (strong negative correlation) to 1 (strong positive correlation). A perfect Spearman correlation of +1 or −1 occurs when each of the variables is a perfect monotone function of the other. A value of zero indicates the absence of association. In Fig. 13 (a) is illustrated the KNIME workflow implementing correlation algorithm. Figure 13 (b) and Fig. 13 (c) show the correlation matrix results. From the matrix is observed a strong correlation between the outlet 3 (Sub_metering_3) and the outlet 1 (Sub_metering_3) and between the outlet 3 and the active power, moreover the reactive power is correlated with the outlet 2 (Sub_metering_2):

according to Fig. 9 (a) the outlet 3 abosrbs the main active energy, besides the loads connected to outled 2 are unbalanced (this will induce to control the electrical network of outlet 2 in order to optimize energy costs and to define the best thresholds).



Rows containing missing values are filtered. Please resolve them with the Missing Value node.

Row ID	Global active power	Global reactive power	Sub metering 1	Sub metering 2	Sub metering 3
Global_active_power	1	0.272	0.339	0.212	0.587
Global_reactive_power	0.272	1	0.126	0.426	0.06
Sub_metering_1	0.339	0.126	1	0.067	0.171
Sub_metering_2	0.212	0.426	0.067	1	0.052
Sub_metering_3	0.587	0.06	0.171	0.052	1

Figure 13. KNIME: correlation matrix results.

In table 3 are summarized the advantages and the disadvantages of the whole model proposed in the paper:

Advantages	Disadvantages
Great reliability level of the electrical power prediction	Priority rules defined on the loads connected on two electrical outlets (for more complex load distribution is required a more complex flowchart implementation)
LSTM model applied on real electrical consumption and exhibiting high performance	//
Possibility to predict load consumption on different windows time (minute, hour, day, week , month, etc.)	//
Possibility to understand in post processing modality the failure/malfunction causes (by means of correlation matrix results)	//

Table 3. Advantages and disadvantages of the features and results of the proposed work.

In the following table are listed all the main parameters and variables adopted in the proposed worked:

Parameters/variables	Description
Sub_metering_1	Energy absorption of the loads expressed in Wh (watt hour of active energy) related to the electrical outlet 1
Sub_metering_2	Energy absorption of the loads expressed in Wh (watt hour of active energy) related to the electrical outlet 2
Sub_metering_3	Energy absorption of the loads expressed in Wh (watt hour of active energy) related to the electrical outlet 3
Global_active_power	Global active power absorbed by all the electrical outlets, expressed in kW (kilowatt)
Global_reactive_power	Global reactive power absorbed by all the electrical outlets, expressed in kW (kilowatt)
Prediction time window	Minute

Table 4. List of parameters and variables used in the model.

The results obtained in the proposed work derive from the comparison and from the application of the model analyzed in [25]. The proposed work is an extension of the work [25].

4. CONCLUSION

The proposed work shows results of an LSTM neural network addressed on logics for enabling and disabling loads of a building. The logics are based on the estimation of loads thresholds, on the global active power prediction and on the comparison with load curve slope prediction. By applying the LSTM algorithm has been observed a good performance of the model predicting global active power. Finally correlation analysis provided important information about the real status of an examined electrical network having loads connected to three intelligent electrical outlets. The intelligence of the prototype system is the controlling of the total electric power by means of consumption prediction, by defining priority rules of loads and of electrical outlets, and by analysing possible unbalanced loads or electrical malfunctions.

ACKNOWLEDGEMENTS

The work has been developed in the frameworks of the industry project: “Sistema MultiPresa/Data Mining Intelligente orientato al Building Electrical Management ‘Intelligent Electrical Outlets’ [Intelligent Electrical Multi Outlets/Data Mining System oriented to Building Electrical Management ‘Intelligent Electrical Outlets’]. The authors would like to thank the following researchers and collaborators: G. Birardi, V. Calati, G. Lonigro, V. Maritati, D. D. Romagno, G. Ronchi, and G. Sicolo.

REFERENCES

[1] Gross, P. et al. (2006) "Predicting Electricity Distribution Feeder Failures Using Machine Learning Susceptibility Analysis", Proceeding IAAI'06 18th conference on Innovative Applications of Artificial Intelligence, Vol. 2, pp 1705-1711.

[2] Zipperer, A., Aloise-Young, P. A., Suryanarayanan, S., Zimmerle, Roche, D. R., Earle, L., Christensen, D., Bauleo, P. (2013) "Electric Energy Management in the Smart Home: Perspectives on Enabling Technologies and Consumer Behavior", Proceeding of the IEEE, Vol. 101, No. 11, pp 2397-2408.

[3] Singh, R. P., Gao, P. X., Lizotte, D. J. (2012) "On Hourly Home Peak Load Prediction", Conference on Smart Grid Communications (SmartGridComm), Proceeding of IEEE Third International Conference on Smart Grid Communications (SmartGridComm).

[4] Mohsenian-Rad, A.-H., Alberto Leon-Garcia, A. (2010) "Optimal Residential Load Control with Price Prediction in Real-Time Electricity Pricing Environments," IEEE Transactions on Smart Grid Vol. 1, No. 2, pp 120 -133.

[5] Vazquez, F. I., Kastner, W., Gaceo, S. C., Reinisch, C. (2011) "Electricity Load Management in Smart Home Control", Proceedings of Building Simulation 2011: 12th Conference of International Building Performance Simulation Association, Sydney, 14-16 November 2011, pp 957-964.

[6] Miceli, R. (2013) "Energy Management and Smart Grids", Energies, Vol. 6, pp 2262-2290.

[7] Agyeman, K., A., Han, S., Han, S. (2015) "Real-Time Recognition Non-Intrusive Electrical Appliance Monitoring Algorithm for a Residential Building Energy Management System", Energies , Vol.2, No. 8, pp 9029-9048.

[8] Aman, S., Frincu, M., Chelmis, C., Noor, M. N., Simmhan, Y., Prasanna, V. (2014) "Empirical Comparison of Prediction Methods for Electricity Consumption Forecasting", University of Southern California, Tech. Rep, pp 14-942.

[9] Callaway, D. S., Hiskens, I. A. (2011) "Achieving Controllability of Electric Loads", Proceedings of the IEEE, Vol. 99, No. 1, pp 184- 199.

[10] Seppala, A. (1996) "Load research and Load Estimation in Electricity Distribution", VTT Publications 289, ISBN 951-38-4947-3, pp 1-118.

[11] Yardi, V. S. (2015) "Design of Smart Home Energy Management System", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 3, No. 3, pp 1851-1857.

[12] Barbato, A., Capone, A., Carello, G., Delfanti, M., Falabretti, D., Merlo, M. (2014) "A framework for home energy management and its experimental validation", Energy Efficiency, Vol. 7, No. 6, pp 1013-1052.

[13] Anastasi, G., Corucci, F., Marcelloni, F. (2011) "An Intelligent System for Electrical Energy Management in Buildings", Proceeding of 11th International Conference on Intelligent Systems Design and Applications (ISDA).

[14] Yan, M. (2012) "The Design and Application of Intelligent Electrical Outlet for Campus's Electricity Saving and Emission Reduction", Journal of Computers, Vol. 7, No. 7, pp 1696-1703.

[15] Jabbar, Z. A., Kawitkar, R. S. (2016) "Implementation of Smart Home Control by Using Low Cost Arduino & Android Design", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 5, No. 2, pp 248-256.

[16] Jing-Min Wang andMing-Ta Yang (2014) "Design a Smart Control Strategy to Implement an Intelligent Energy Safety and Management System", Hindawi Publishing Corporation: International Journal of Distributed Sensor Networks, Vol. 2014, No. 312392, pp 1-10.

[17] Iwayemi, A., Wan, W., Zhou, C. (2011) "Energy Management for Intelligent Buildings", Energy Management Systems, Edited by Dr Giridhar, Kini, INTECH book 2011.

[18] Dent, I., Aickelin, U., Rodden, T. (2011) "The Application of a Data Mining Framework to Energy Usage Profiling in Domestic Residences using UK Data", http://dx.doi.org/10.2139/ssrn.2829282 .

[19] Gajowniczek, K., Ząbkowski, T. (2015) "Data Mining Techniques for Detecting Household Characteristics Based on Smart Meter Data", Energies, Vol. 8, No. 7, pp 7407-7427.

[20] Fernández-Caramés, T. M. (2015) "An Intelligent Power Outlet System for the Smart Home of the Internet of Things", Hindawi Publishing Corporation: International Journal of Distributed Sensor Networks, Vol. 2015, No. 214805, pp 1-11.

[21] Mocanu, E., Nguyen P. H., Gibescu, M., Kling, W. K. (2016) "Deep Learning For Estimating Building Energy Consumption", Sustainable Energy, Grids and Networks, Vol. 6, pp 91-99.

[22] Zheng, J., Xu, C., Zhang, Z., Li, X. (2017) "Electric Load Forecasting in Smart Grids using Long-Short-Term-Memory based Recurrent Neural Network", 1st Annual Conference on Information Sciences and Systems (CISS), pp 1-6.

[23] Okafor, K. C., Ononiwu, G. C., Precious, U., Godis, A. C. (2017) "Development of Arduino Based IoT Metering System for On-Demand Energy Monitoring", International Journal of Arduino based IoT Metering System for On-Demand Energy Monitoring, Vol. 7, No. 23, pp 3208-3224.

[24] Folayan G. B., Idowu, O. O. (2018) "Remote Controlled Advanced Power Strip Using Raspberry Pi for Effective Energy Management", Innovation Energy & Research, Vol. 7, No. 1, pp 1-4.

[25] Massaro, A., Galiano, A., Meuli, G. Massari, S. F. (2018) "Overview and Application of Enabling Technologies oriented on Energy Routing Monitoring, on Network Installation and on Predictive Maintenance" International Journal of Artificial Intelligence and Applications (IJAIA), Vol.9, No.2, pp 1-20.

[26] "Individual household electric power consumption Data Set" 2018. [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Individual+household+electric+power+consumption

[27] Martino, L. M., Amarasinghe, K., Manic, M. (2016) "Building Energy Load Forecasting using Deep Neural Networks" IEEE Proceeding of the 42nd Annual Conference of the IEEE Industrial Electronics Society (IECON).

[28] "Kaggle" 2018. [Online]. Available: https://www.kaggle.com/

[29] Myers, J. L., Well, A. D. (2003) "Research Design and Statistical Analysis", (2nd ed.) Lawrence Erlbaum. pp. 508, ISBN 0-8058-4037-0.

CorrespondingAuthor

Alessandro Massaro: Research & Development Chief of Dyrecta Lab s.r.l.