CIGRE-325 **2020 CIGRE Canada Conference**
Toronto, Ontario, October 19-22, 2020

Energy consumption estimation through Artificial Intelligence

Adrián Luis Paolini, Gustavo Serrano
Edenor S.A.
Argentina

SUMMARY

Smart energy meters are still a novelty for a large portion of the world. And we, who don't have yet the means to acquire them, must sharpen our wit to comply with today's necessity for instantaneous, precise information.

Given our almost 3 million customer base, every process or indicator concerning them must be fully automated or out rightly simplified, with the possible consequence of a substantial loss of precision.

Our company needs to estimate its total energy sold every month and to be more precisely at the end of the month, of course we don't readily have that data, as a cycle of meter readings takes about two months. So, for now we are using the simple method of getting the daily consumption of the previous period, and applying the value to the days passed of the current period. This method is very imprecise, as the patterns of consumption of a bimonth can't be applied to the next one.

In light of this problem, we speculated that in theory, we could use artificial intelligence to break down which factors make every single customer consume energy, and replicate them on different situations. We just needed to find the right neural network and find out the factors that could influence the customers' behaviour.

KEYWORDS

Forecasting, Deep Learning, Artificial Intelligence, Power Dispatching, Electrical Distribution.

apaolini@edenor.com

About us

We work for the biggest electrical distribution company in Argentina, with approximately 3.5 million clients, with 99% of them having electromechanical meters or meters with no communications.

About our project

The Financial Direction of our company needs to know the total energy sale on a monthly basis, but as most of our meter system is electromechanical, the energy sale estimation is something difficult to calculate, especially when each complete loop of meter readings is made along two months.

In order to get some estimation, it is quite common to linearize the problem and make the estimation just by multiplying the daily energy of the previous period, times the days passed in the current month. This method tends to it calculates a bigger value in autumn and spring, and a smaller value in summer and winter.

Given our recent discovery of artificial neural networks (ANN), we found out that this could make a typical textbook case about the use of the technique, as with it we could try to learn every customer's behaviour, and emulate them later. We just needed to find the right set of parameters on which our customers base their energy consumption habits.

Getting started with the task

Finding the right model of neural network

We got a database of our customers' energy usage from 2007 to 2018, with an intention to contrast our calculated values to the real ones. Also, for starters, we got hourly information about the weather conditions in that same period.

After trying with several types of ANN, we ended up adopting a recurrent neural network (RNN), as we concluded that the progression of values was a very important factor for predicting the following one.

Our RNN of choice was finally the called Long-Short Term Memory, or LSTM for short. This type of neural network has the ability to not only remember past calculations, but as well to calculate which parameters are the most relevant to remember and for how long. We found this one very suitable for our case of use because we know that a household's behaviour is consistent along the time, but also that it can change its habits or its situation. For example, acquiring new electrical appliances, family members moving in or out, etc.

Identifying the key inputs

Early on, we divided the hours of the day in three different groups to separate business, leisure and sleep hours. Thus, every time-depending parameter would generate three different inputs for the ANN.

The first and most basic parameter in the network is the amount of days in the period of analysis. This parameter impact on a linear way the final value. Soon after that, we decided to put the amount of working days of the period, as we understand that in these days, the customer would have a completely different behaviour compared to the non-working days. So, there were the two first inputs of our ANN.

One of the most important parameter to take account seems to be the climate temperature, because people tend to turn on devices to try to make their houses more comfortable. We didn't take just the average temperature of the period, as it would mask the details of every passing day. For example, a very hot day would nullify a very cold day, and the ANN could assume that none of the days had a very large energy expenditure when, in reality, both days had.

Instead, we added the absolute value of the difference between each hourly climate temperature and the comfort temperature, set for now at 18º C. After some tests, we decided to generate two separate parameters: one for all the temperatures above the comfort, and another one for all those below the comfort *(fig. 1)*. This allowed us to better discern the nature of the electrical devices for temperature control of each household. This factor resulted in six different inputs for the ANN.

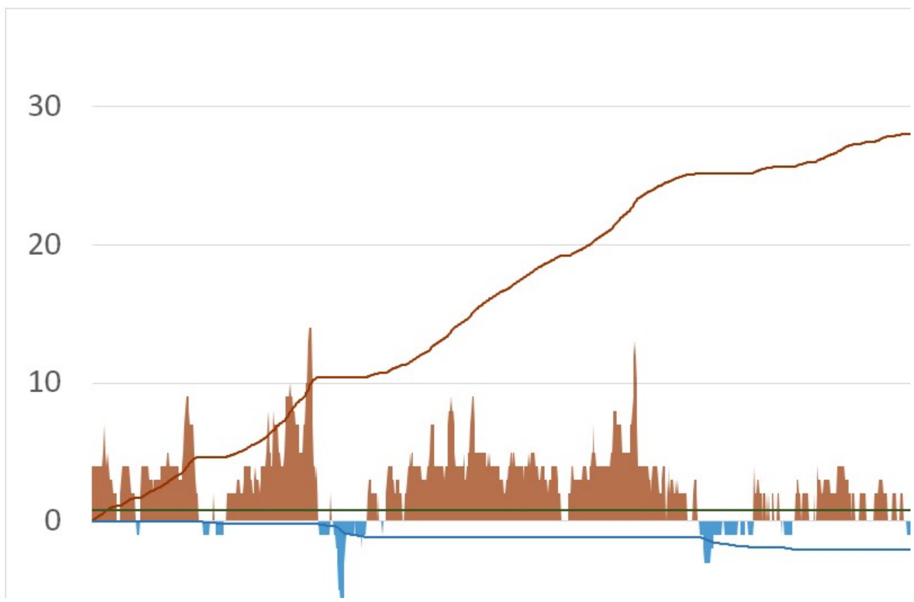

fig. 1: integrated superior and inferior temperatures versus average temperature.

Our next parameter was about the lighting conditions. It is expected that by night or an overcast day, the customer will turn on more lights than on a bright day. As in the previous parameter, we accounted a luminosity factor for every hour of the period, ranging from 0 representing total darkness to 1 representing very bright conditions *(fig. 2a and 2b)*. Of course, this factor shouldn't mean the same on a cloudy afternoon and after midnight, when everybody is sleeping, so dividing this parameter in the three times of the day was very important. This component added three more inputs for the network.

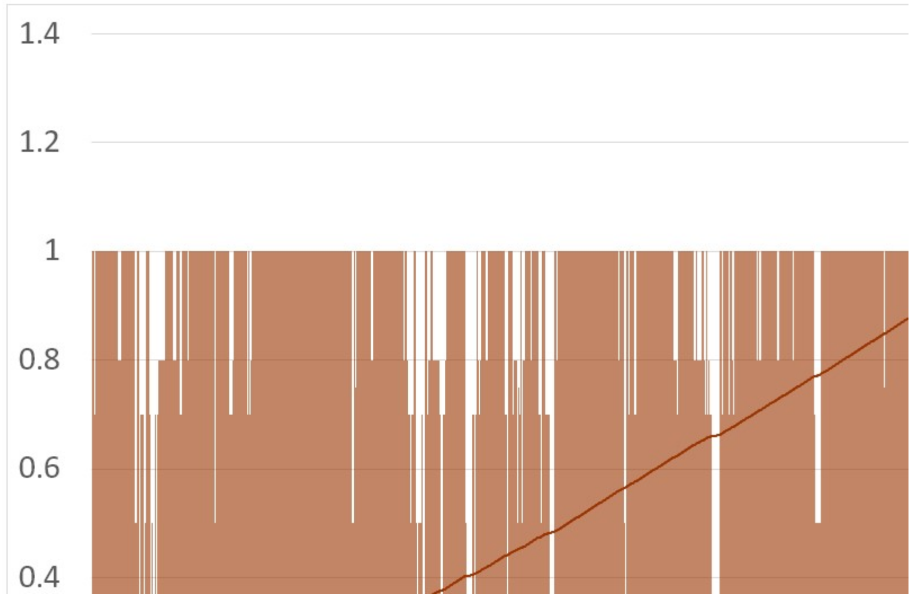

fig. 2a: lighting factor in summer and its integrated value

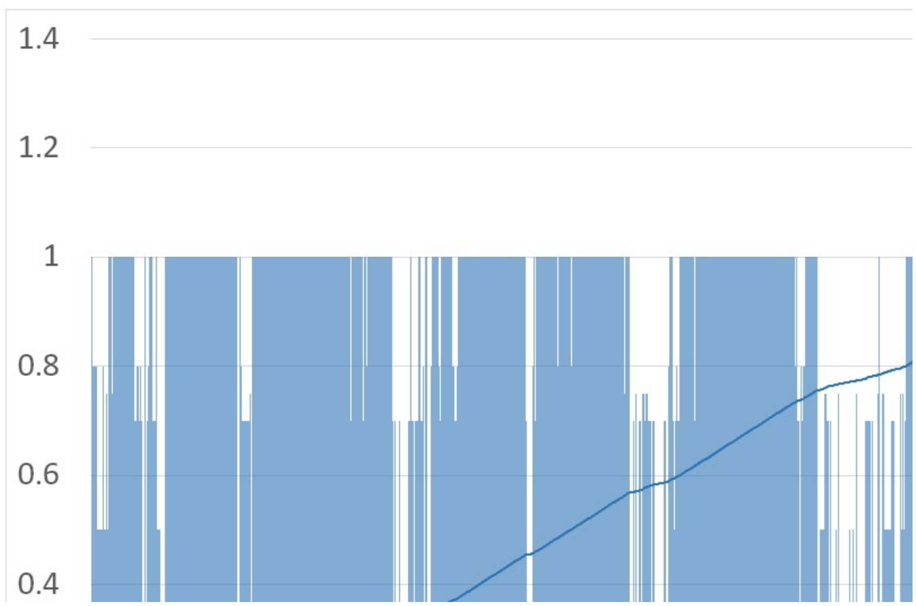

fig. 2b: lighting factor in winter and its integrated value

Lastly, we added the average daily energy consumed in the previous period. This last input proven to be of the upmost importance, because without it results diverged too much.

So, after multiple sets of tests to model the network, we ended with an LSTM network of twelve inputs.

Technical information about the neural network

As we mentioned before, we developed an LSTM network [1, 2]. To generate it, we used Python with the TensorFlow library. We set the network with the following parameters:

Paper Cont'd

-Activation function: hyperbolic tangent
-Dropout: 0.3 [3]
-Loss: mean square error
-Optimizer: RMSprop
-Epoch: 250
-Batch size: 1

It has only one internal layer of 16 dense neurons and one neuron layer as way out. We tried to use more internal layers but the system got overfit most of the times.

Getting the ANN to work

The first trial was applied to a set of 88 customers. We selected only residential customers because it's relatively safe to assume that their meters won't be replaced in a short term.

As we said before, we counted with the energy spent by the customers from 2007 to 2018. We cycled the process training the network with 7 bimesters and asking it to yield the energy consumption of the following one taking in to account its weather conditions. We repeated this loop until we ran out of bimesters to estimate.

Our personal workstation took 25 minutes to estimate 57 periods of 10 years for the 88 customers. At that rate, it would take us way longer than a month to analize each customer of the company. Nevertheless, we calculate that using multiple dedicated servers with GPU capabilities, this process would take only a couple of days.

After this, we ended with 88 different neural networks, each one of them trained to try to replicate a particular customer's behaviour for potentially every type of weather condition.

Analysing the results

When we contrasted the results of our method against the previous one, we noticed that the phasing shift problem had gone, and that the sum of our estimations were highly similar to the sum of the real values *(fig.3)*.

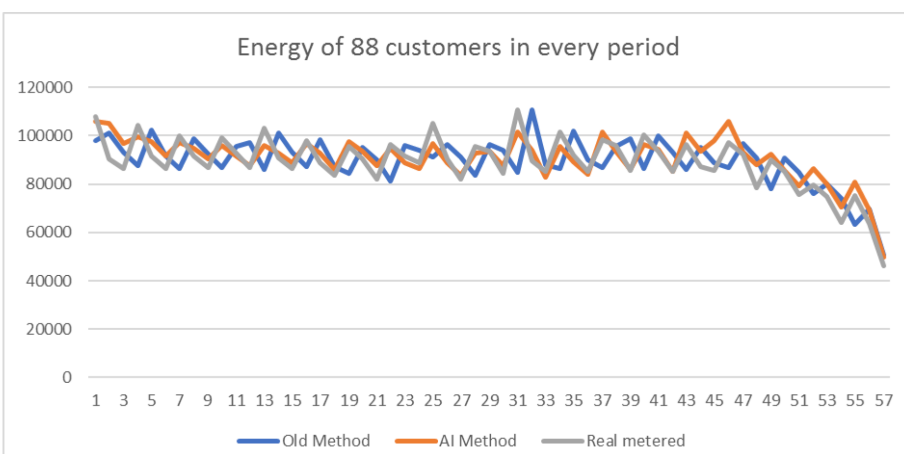

fig. 3: comparison of both methods. The drop of values to the right responds to a lower demand of energy, completely independent of this exercise.

Yet, as our method is much more complicated that the former, it could present some problems yielding a coherent result. For example, as we already expected, this method works best on customers who have a very regular routine *(fig. 4)*, but could easily return an absurd value on those customers who have a sharp change of behaviour *(fig. 5)*. Of course, the ANN itself would heal once it's trained with this new routine.

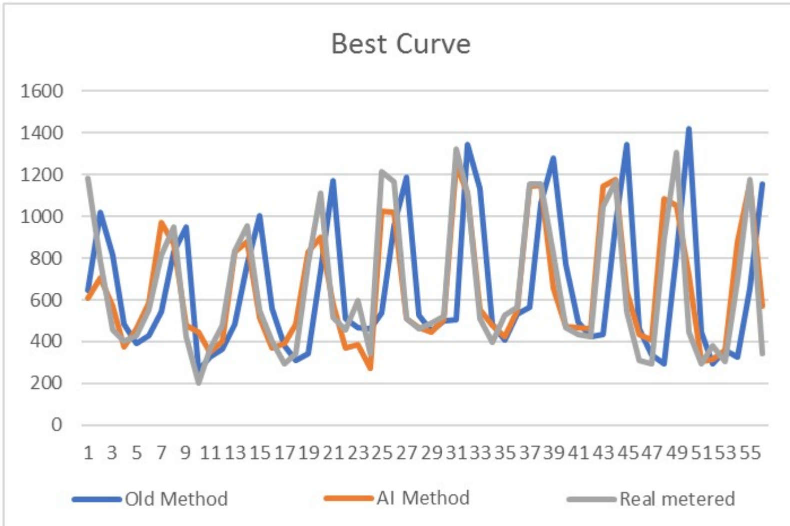

fig. 4: the most regular customer of the set

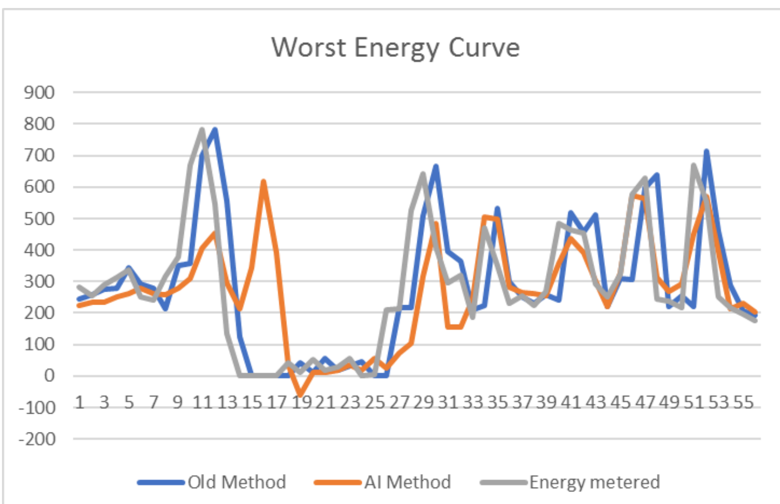

fig. 5: this customer has a break on their habits, and so the ANN returns a couple of very inaccurate estimations, but it quicky adapts to the new conditions.

However, those nonsensical values could prove to be useful, as in our business, a sudden change of energy consumption could mean that the customer is tampering with its meter. So, even if it wasn't our original goal, this method could help our company to reduce meter related frauds.

Another use of this method could be to deduce if each customer uses electrical energy to cool or heat their house. This information should be valuable for our marketing department.

Lastly, knowing our entire customer base habits, we could run simulations on how would our electrical network endure under certain weather conditions. For example, we could see if any part of the network would collapse by having a heat wave of 33ºC for 10 days.

In conclusion

From the start of the project, we saw this methodology of getting inside our customers' heads as a novelty, for none of us had before seen such a level of granularity in any company-wide customers' analysis.

Furthermore, when we started, our company had yet to implement a big data department. So, we were kind of pioneering the technique. We found out that the use of neural networks lets us unravel invisible patterns, and we are already using them for a couple of new projects.

We foresee that, as processing power and cheap software alternatives evolve, artificial intelligence is becoming less of an exclusivity of IT mega corporations, and more of a tool for medium to big enterprises to help them deal with their regular issues.

End of text

BIBLIOGRAPHY

[1] Keras LSTM tutorial – How to easily build a powerful deep learning language model, https://adventuresinmachinelearning.com/keras-lstm-tutorial/
[2] Keras, LSTM layer, keras.io/api/layers/recurrent_layers/lstm/
[3] Machinecurve, How to use Dropout with Keras? www.machinecurve.com/index.php/2019/12/18/how-to-use-dropout-with-keras/