Luis Vizcaya

# Energy Market Forecasting Challenge

## The problem

Given a dataset with the system load, system solar forecast, planned forecast, and system wind forecast, create a model to jointly predict scenarios for real time, day ahead lmps, and wind generation.

## The approach

There are several alternatives to tackle the forecast scenarios challenge. In this case, it was faced through a multi-layer perceptron (MLP), with one input layer with four input variables, three hidden layers and one output layer with the three target variables

The multi-layer perceptron has characteristics that make it convenient for this challenge.

- Capture complex relationships between input and output variables. The MLP contains ReLU layers that let the model learn non-linear relationships.

- Manage joint predictions intrinsically, as the loss function is calculated from the error of all the target variables during the training process, the model learns the cross-correlations between the output variables.

- Can behave as an ensemble model, as a mix of neural networks, if it is configured for that purpose.

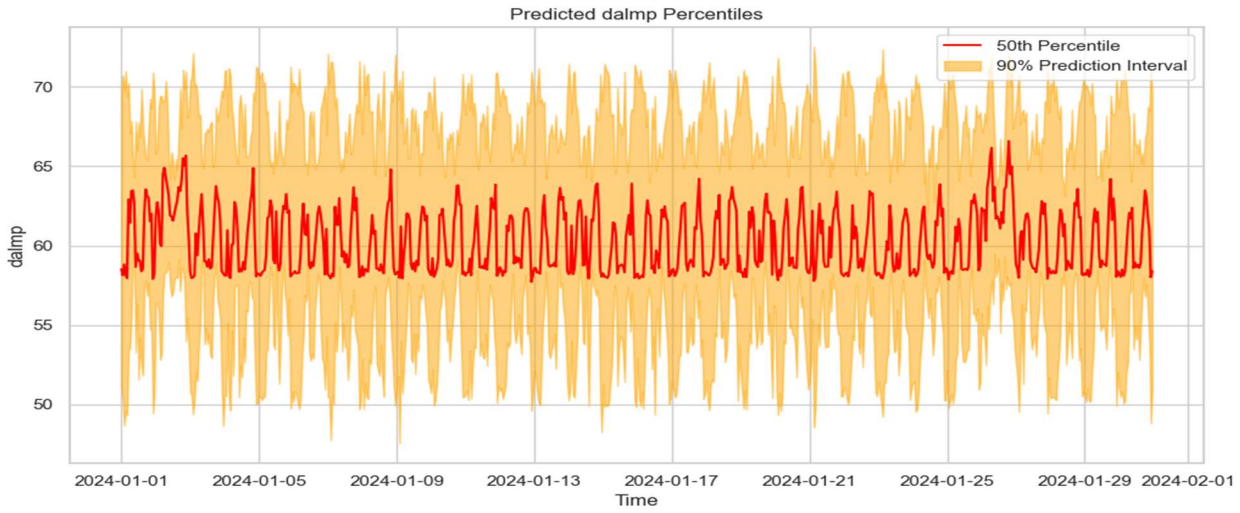
- Models are easy to train and tune (learning rate, hidden layer, number of epoch), compared with more complex models (GAN, Transformers)

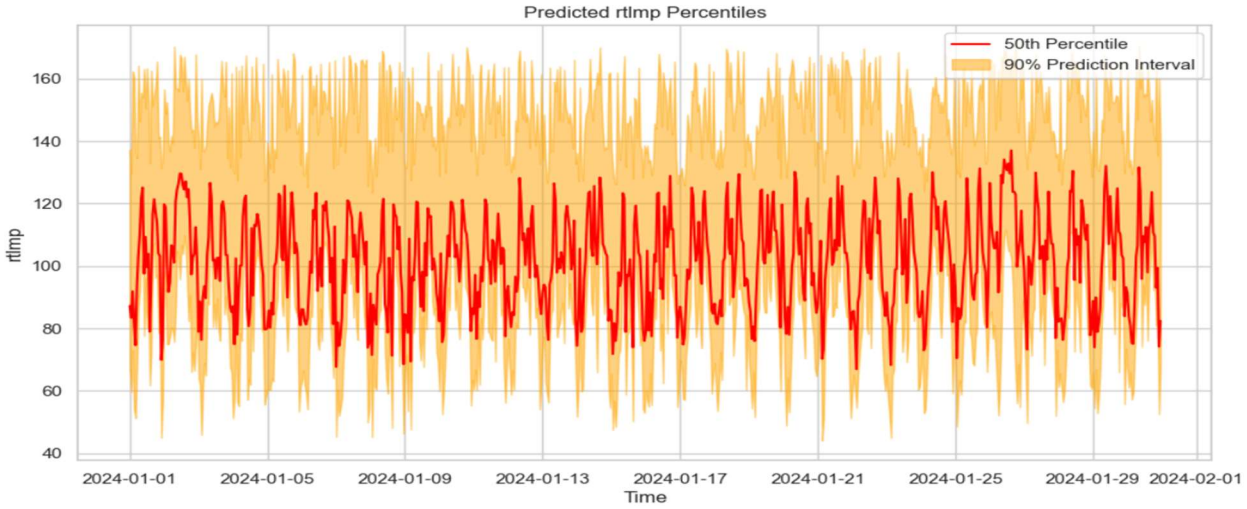
## Neural Network as ensemble model

- Neural networks can behave as ensemble models, if the dropout layers are activated during inference, and neurons are dropped randomly (Monte Carlo Dropout). For every run, the topology of the network will be different, and a different prediction will be generated. That was the case implemented for generating different scenarios for each hour in the test set.

## Results

The graphs below show the central tendency (50% percentile) and the uncertainty (90% prediction interval) for real time, day ahead prices for the test set (Jan-2024).




**Scaling and Productize**

- The model can be deployed as a web API, specifically the class predictor can be wrapped into a FastAPI endpoint. Web Apis is the preferred method used by customers to integrate our prediction services into their system.

- For productizing, an ETL pipeline can be coded and implemented, for processing data automatically, ensuring.

- The training process can be scheduled to run weekly to keep the model update with the last market dynamic.

- Deploying the script on a cloud service (AWS, GCP or Azure), it provides scalable processing power on demand, and tools for training models, monitoring forecast accuracy and alerting.

- The pipeline can be run in a docker container on a serverless container service, for ensuring its reproductivity and resilience regardless of the operating system and scale on demand.

- A database can be implemented for storing the historical predictions, and input variables as well.