POLITECNICO MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE

EXECUTIVE SUMMARY OF THE THESIS

Voltage regulation in distribution networks in the presence of distributed generation and electric vehicles: LVR and E-OLTC with machine learning approach

TESI MAGISTRALE IN ELECTRICAL ENGINEERING – INGEGNERIA ELETTRICA

AUTHOR: DURIM MUSIQI

ADVISOR: DURIM MUSIQI

ACADEMIC YEAR: 2020-2021

Abstract – This thesis deals with voltage regulation in MV networks with high penetration of distributed generation (DG) and electric vehicles (EVs). Line voltage regulators (LVRs) and Electronic On-Load Tap Changers (E-OLTCs) are used exclusively as voltage regulation methods without help from any reactive power control mechanisms. This thesis will make use of machine learning algorithms, more specifically Deep Neural Networks (DNNs) to solve the problem of voltage regulation in these networks. The methodology used here is to model two MV feeders, create and assign 2-year characteristics to each load and to each photovoltaic plant. Then, Quasi-Dynamic (Q-D) simulations of 2-years with hourly steps will be run. From the Q-D analysis, the results of tap changer's positions will be saved and used to train the deep neural networks. The PVs are modelled to have a constant power factor of 0.95. Their active power characteristic is built using solar irradiation database SARAH2 from European Commission's PVGIs tool. The characteristics of loads are created using a similar feeder's load profile for a year. The first goal of this study is to evaluate solely the impact of automatic tap changers (ATPs) in the feeder, hence not utilizing other methods. The second goal is to get an initial understanding of how machine learning can predict the correct tap positions of transformers when given inputs of loads and PV outputs.

1. Introduction

In classical MV networks the power used to flow downstream, from the substations, along the lines, and towards the loads. This has been the case for more than 100 years, however, networks started to change after intense research in renewable energy sources (RES) was followed by a steep decline in prices of photovoltaic panels. The price dropped from 105 [$/Watt] in 1975 to just 0.2 [$/Watt] [1].
Governmental incentives, public interest and common international targets for renewable sources pushed for medium networks to host even larger amounts of distributed generation (DG). One such target is the EU Directive 2018/2001 [2]. DGs have a positive impact in general, as they lower CO_2 emissions, however, the downside is that RES render the power system more difficult to operate. That happens because of their unpredictable nature [3]. Even if the predictions of irradiation and wind velocity are correct, DGs may cause voltage fluctuations, over-voltages and even flicker [4].
For each load and PV system in the network, 2-year hourly time-series are used. That translates to 17520 data points for each characteristic. For the loads, both the active and reactive powers will have their own characteristics. As for the PVs, both the active and reactive power they produce will also have their own characteristics. This thesis contains two studycases which will consider two different feeders. One is sub-urban and operates at 20 [kV] while the other is rural and operates at 10 [kV]. Both feeders are modelled from scratch in PowerFactory, using IEEE 33 bus as a base for the first one, while the second one is a real representation of feeder Rugova in Kosovo. In the first case, the installed peak power of the photovoltaic systems is around 72% of the aggregate peak loads of the feeder, while on the second case, it is around 90%. All transformers in MV are considered to have 5 tap positions including the neutral, with an additional voltage of 2.5% per tap.

The topologies of the feeders are depicted in Figure 1 and Figure 2.

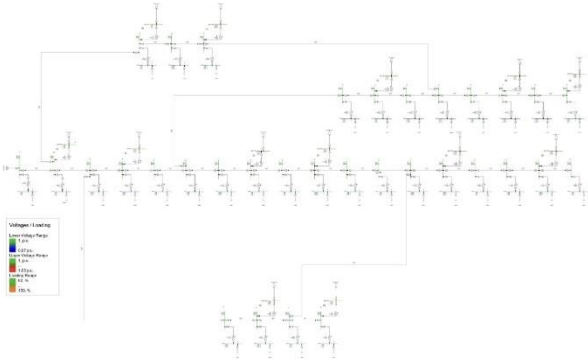

Figure 1. Feeder 1 20 [kV] – SubUrb.

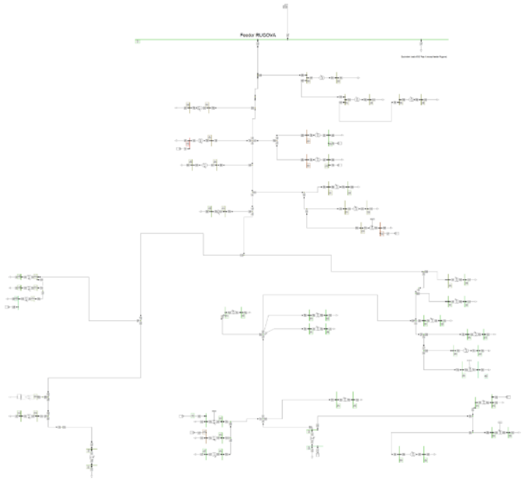

Figure 2. Feeder 2 10 [kV] – Rugova.

2. LVRs vs E-OLTCs

Both LVRs and E-OLTCs are devices that make it possible to adjust the voltage, usually in a range of ±5% or ±10% of the nominal voltage.
LVRs are suitable for longitudinal feeders, where they are preferred to be installed somewhere in the middle of the main branch. In these cases, LVRs are able to regulate the voltage for a portion of the feeder. Figure 3 shows a typical longitudinal feeder where LVRs are most effective.

Figure 3. Prizren-Zhupa feeder in the Sharr Mountains.

In feeders similar to Feeder Zhupa which is around 35 [km] long, one or more LVRs installed in the middle of the main branch will be effective in both compensating voltage drops when the load is large, and also in decreasing voltages at times when DGs cause over-voltages due to high generation and small loads.
However, LVRs have not shown plausible results often enough when dealing with urban or sub-urban feeders, especially when the number of DG points is high. That happens because those feeders are much more branched and shorter relative to rural feeders. This conclusion was reached after several tests in a longitudinal feeder where a LVR is installed somewhere in the middle of the feeder.

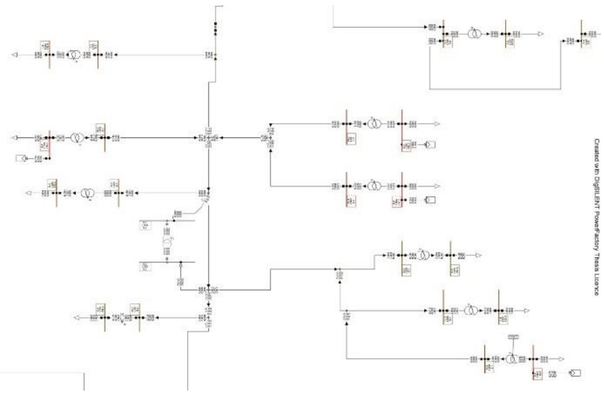

Figure 4. Feeder with high DG facing overvoltages: LVR not employed

In Figure 4 the feeder experiences overvoltages (red busbars) due to large amounts of distributed generation. In this case, LVR is bypassed and out of service.

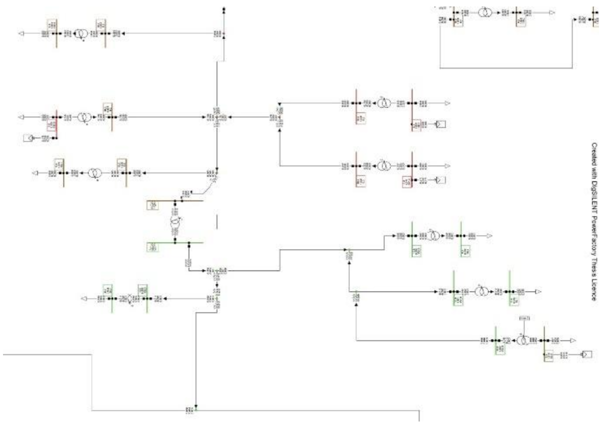

Figure 5. Feeder with large DG facing overvoltages: LVR employed.

In the second case, shown in Figure 5, LVR is put to service, and though it is able to lower the voltage on one side of the feeder, it cannot help with overvoltages in busbars situated before the LVR. After performing several similar tests, with LVR being installed at various points in the network, it was deemed that LVRs aren't an optimal solution for voltage regulation in networks with many long branches. Installing several LVRs at once is another possibility, but that may not be feasible compared to simply having transformers where PVs inject equipped with E-OLTCs. For this reason, in this study, E-OLTCs are the main tool taken into consideration for overvoltages. Anyhow, LVRs are used mainly in combination with E-OLTCs to help augment undervoltages at times of large aggregate load and low or no DG injection.

3. Preparing the characteristics of loads

It is rather difficult to obtain real data from distribution companies, therefore, the characteristics of loads (active component) are based on a single dataset provided by a German DSO. This characteristic (**Lr**) belongs to a similar feeder with household users as the ones in our studycases.

$$\mathbf{Lr} = (L_{h1}, L_{h2}, L_{h3}, \dots, L_{h17520})_{1x17520}$$

The loads characteristic is first scaled from zero to one to create a vector (**Ln**) that contains 2-year hourly data of the loads.

$$\mathbf{Ln} = \mathbf{Lr}/\mathrm{Lrmax}$$

Then, it is multiplied to the vector of peak values of all loads in the feeder (**Lp**), transposed.

$$\mathbf{Lp} = (L_{p1}, L_{p2}, L_{p3}, \dots, L_{pN})_{1xN}$$

$$\mathbf{L} = (\mathbf{Ln}' \times \mathbf{Lp})_{17520xN}$$

Matrix **L** contains *N* columns, where *N* is the number of loads in the feeder. Each column represents the 2-year hourly characteristic of the respective load. As for the reactive component of loads, a similar matrix of size 17520×N can be created utilizing power factors of each load, which are given. The idea of this study is to try different combinations of loads and PV inputs, so the load characteristic is the same in both years in the account of the PV characteristic changing in the second year, which is enough to fish out valuable results. The characteristic of a random load in one of the studycases is shown in Figure 6, while Same goes for all other loads.

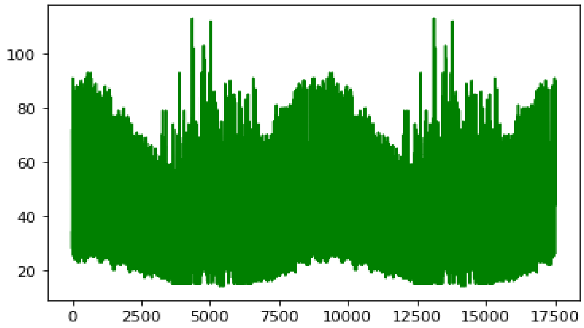

Figure 6. Characteristic of Load 12 in Feeder 1 [kW].

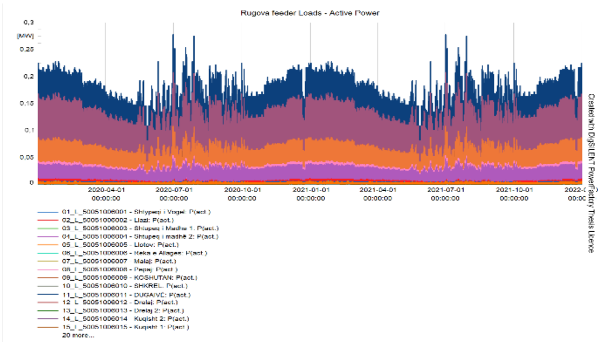

Figure 7. All loads of Feeder Rugova [kW].

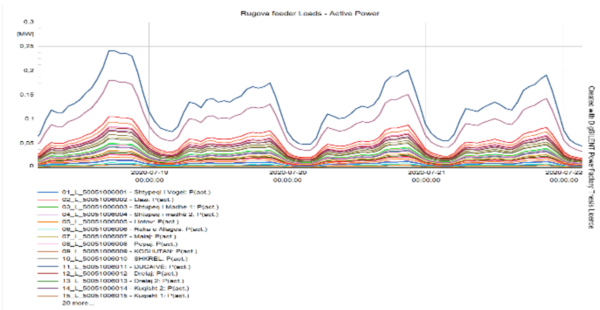

Figure 8. All loads of Feeder Rugova for a random short period [kW].

4. Preparing the characteristics of PV outputs

As for the PV outputs, the data obtaining methodology used here is identical to the industrial one. The coordinates of the geographical location of the feeder are used to access solar irradiation measurements databases. In this particular case, European Commission's PVGIS tool from SARAH2 solar radiation database is utilized. SARAH2 offers data on Direct Normal Irradiance data (DNI), Diffuse Horizontal Irradiation (DHI), and Global Horizontal Irradiance (GHI), however, it also offers data directly in terms of power, measured in kilowatt.
Since there are PVs with different sizes (installed peak power), first the power output of a PV plant of 1[kW] is downloaded and made into a vector **Ipv.**

$$\mathbf{Ipv} = (0, 0, 0, \dots, 0.89, 0.91, 0.93, 0.88, \dots)_{1 \times 17520}$$

Next, a vector of peak installed powers of PV is built.

$$\mathbf{Ppv} = (\mathrm{PP1}, \mathrm{PP2}, \dots, \mathrm{PPM})_{1xM}$$

where M is the number of PVs installed in the feeder. Now, multiplying these vectors will give a matrix containing the PV output characteristics for all the plants.

$$\mathbf{Po} = (\mathbf{Ipv}' \times \mathbf{Ppv})_{17520xM}$$

Matrix **Po** has M columns, each representing the characteristic of its respective PV plant in the feeder. One such characteristic of hourly steps of 2 years is shown below in Figure 9.

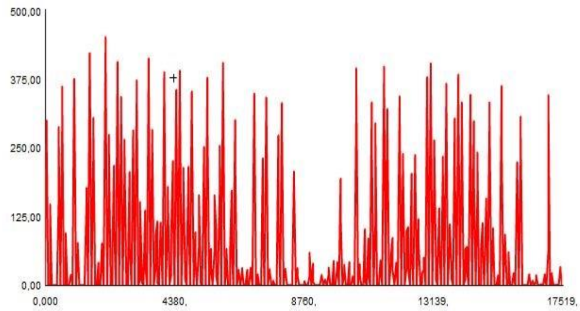

Figure 9. A random PV characteristic [kW].

As the figure shows, the PV output characteristic changes from the first year to the second, thus assuring that the overall inputs will never be the same in any two different time steps, although the load may not change between the first and the second year. The same process is done with all PVs of the feeder. As for their reactive power component, a constant power factor of 0.95 is considered, which coincides with the minimum required level in many countries. Inverter producers guarantee a power factor of up to 0.99, however, the idea here is to see the network's behavior in worst cases.
The next figure shows all PV outputs of one of the feeders, zoomed in a random short period.

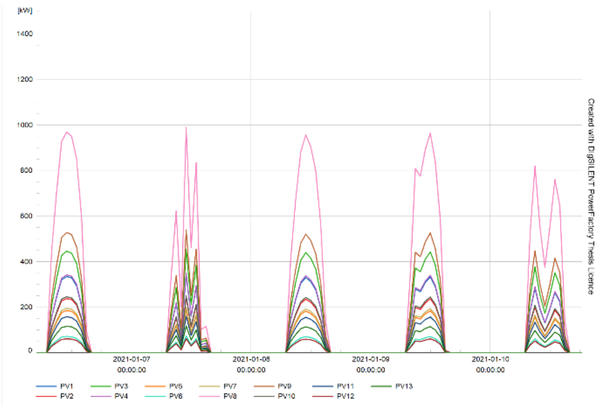

Figure 10. PV outputs of Feeder 1, zoomed in a short random period [kW].

5. Running the Quasi-Dynamic Simulation

In order to train a machine learning algorithm, large training examples must be used. In this study, Q-D simulations were run in the feeders modelled in PowerFactory in order to obtain thousands of training examples. The simulations cover a time-series window of 2 years, in hourly steps. That translate to 17520 load flows. For each load flow that is run, different inputs of loads and PV generation is used, and the results of the correct tap changers are saved. In order to realize the first goal of this study, which is analyzing the effects of automatic tap changers (ATP) in the network, a first Q-D analysis is run without employing ATPs. When that is done, the voltage constraint violations are counted. In the second run of the Q-D analysis, ATPs are employed and the voltage constraint violations are counted again.
Figure 11 shows the tap positions of transformers in a short period of the simulation. The effect of ATP is imminent, as the positions change very often in order to adjust the voltage within the suitable preset range. Feeder 1 shows similar behaviour. In order to obtain as much data as possible, the Q-D simulations are run without including any outages or maintenance. In the future, upon finding relevant data of multiple years, outages and maintenance could be included for more realistic results.

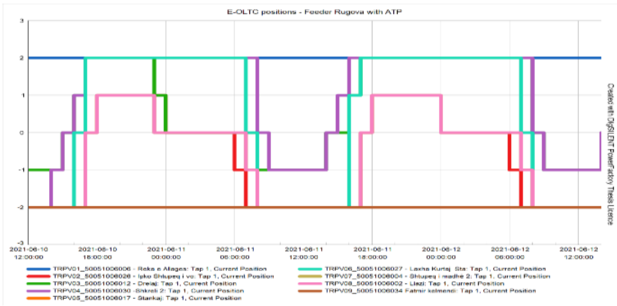

Figure 11. Tap positions of transformers in Rugova feeder when ATP is on.

The results of both feeders show clear improvements of parameters in the case where ATPs are employed.

The correct tap positions are saved from the Q-D analysis with ATPs on. They are later to be used to train the machine learning algorithm.

In numerical terms, as seen in Figure 12, for the first feeder (Sub-Urb) without using ATPs, there were 21,924 instances of voltage constraint violations of over 1.05 [p.u] and no instances of violations of over 1.1 [p.u].

```
SU_noEOLTC_violations_5percent =
(SU_voltages_noEOLTC>1.05).sum().sum()
print(SU_noEOLTC_violations_5percent)
21924
```

```
SU_noEOLTC_violations_10percent =
(SU_voltages_noEOLTC>1.1).sum().sum()
print(SU_noEOLTC_violations_10percent)
0
```

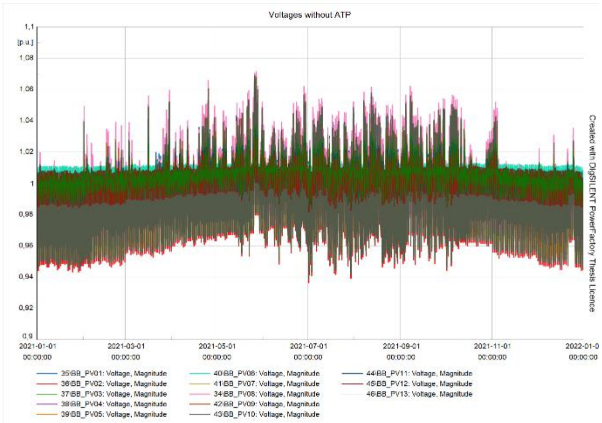

Figure 12. Feeder 1 voltages without ATPs [p.u].

While when ATPs were applied, all voltage constraint violations were eliminated.

```
SU_withEOLTC_violations_5percent =
(SU_voltages_withEOLTC>1.05).sum().sum()
```

```
print(SU_withEOLTC_violations_5percent)
0

SU_withEOLTC_violations_10percent =
(SU_voltages_withEOLTC>1.1).sum().sum()
print(SU_withEOLTC_violations_10percent)
0
```

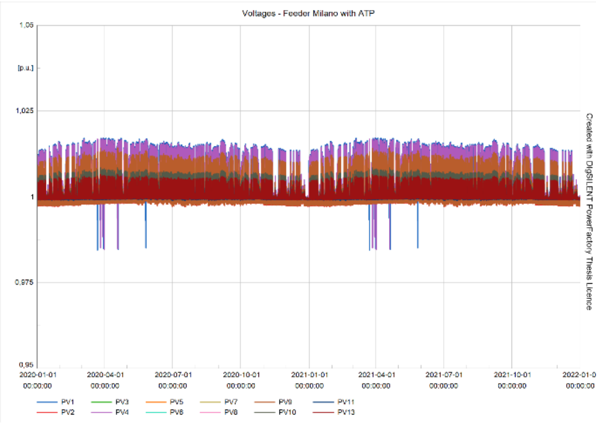

Figure 13. Feeder 1 voltages with ATPs [p.u].

As for the second feeder (Rugova), when ATPs are used, there are 11,174 instances of voltage violations of 5%, and 2,404 instances of overvoltages of 10%.

```
ru_noEOLTC_violations_5percent =
(RU_voltages_noEOLTC>1.05).sum().sum()
print(ru_noEOLTC_violations_5percent)
11174

ru_noEOLTC_violations_10percent =
(RU_voltages_noEOLTC>1.1).sum().sum()
print(ru_noEOLTC_violations_10percent)
2404
```

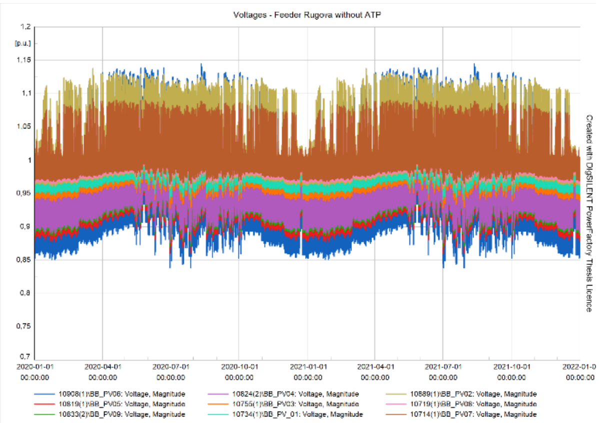

Figure 14. Feeder 2 voltages without ATPs [p.u].

When applying ATPs to the second feeder, there were only 2,432 overvoltage instances of over 5%, and zero instances of voltages over 10%.

```
ru_withEOLTC_violations_5percent =
(RU_voltages_withEOLTC>1.05).sum().sum()
print(ru_withEOLTC_violations_5percent)
2432

ru_withEOLTC_violations_10percent =
(RU_voltages_withEOLTC>1.1).sum().sum()
print(ru_withEOLTC_violations_10percent)
0
```

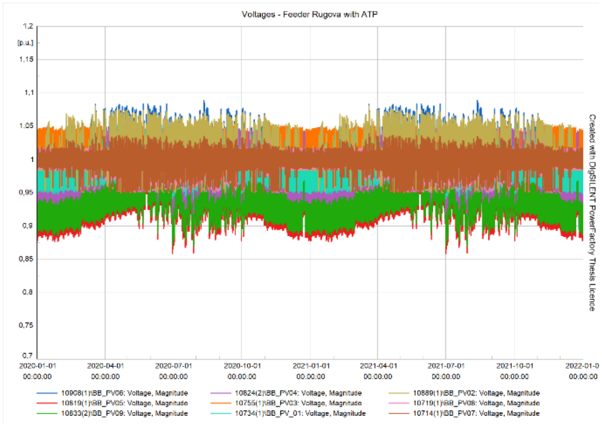

Figure 15. Feeder 2 voltages with ATPs [p.u].

That translates to around 80% less violations of over 5%, and complete elimination of overvoltage violations of over 10%.

At this point, the impact of E-OLTCs in the feeder is clear as they prove to be a very successful tool against overvoltages. However, Figure 15 shows voltages dropping down to 0.85 [p.u], meaning that E-OLTCs do not also guarantee to regulate undervoltages. This is where LVRs come in handy. Feeder 2 will be re-modelled with four additional LVRs installed at strategic points in the feeder, designed specifically to augment undervoltages.

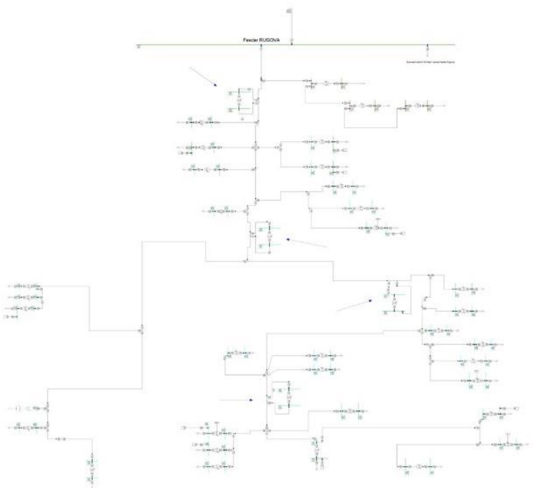

Figure 16. Feeder 2 with four LVRs installed.

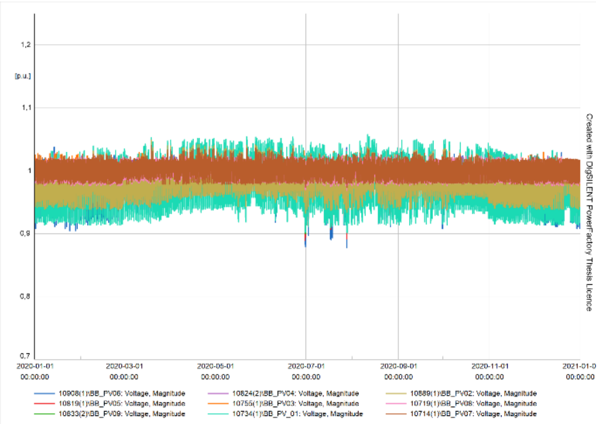

Figure 17. Feeder 2 voltages with E-OLTCs and LVRs combined [p.u].

Figure 17 shows improvements in undervoltages where they are almost 100% avoided. That is thanks to LVRs.

6. Building and training the Deep Neural Networks

The Q-D simulations have provided 17520 training examples for each studycase. The inputs or features used are:

- Active powers of loads
- Reactive powers of loads
- Active power injection of PVs
- Reactive power injection of PVs

The outputs of features are:

- Tap positions of transformers
- Tap positions of LVRs (if they're installed)

With that being said, the DNN model will train in a manner such as when, in the future, it is fed new, unseen inputs as the ones above, it will be able to predict the results of the correct tap positions of all transformers.

As far as the DNN network for this particular problem is concerned, it is deemed feasible that:

i) DNNs should be Multi-Output.

ii) The algorithm should be Non-exclusive.

iii) Classification should be Binary, using Sigmoid activation function in the output layer.

iv) The number of neurons in the output layer will be $5 \cdot N_{transformers}$.

The DNN architecture in discussion will look something like this:

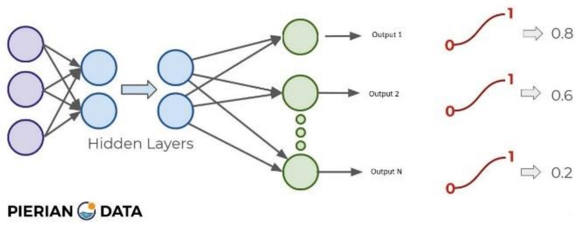

Figure 18. Deep Neural Networks for voltage regulation using E-OLTCs

So, each neuron in the output will represent the probability of its corresponding tap position of a transformer to be the correct one. For example, neuron number 8 represents tap position (0 or neutral) of transformer 2. If that neuron shows a result of 1, it means that given those inputs, the model is predicting that transformer 2 is operating at tap position (0). Since the algorithm in non-exclusive, more than one tap position can be correct simultaneously, which reflects the real life case. The code for both studycases is standard and unchanged, except for the architecture of DNNs, where the number of neurons must be suitable to the specific dataset. Next, the full code of the **second studycase** is shown and the main steps are explained briefly.

First the relevant libraries are imported, then the features and labels.

```
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
X = pd.read_csv('X.csv') #importing the
features (inputs)
y = pd.read_csv('y.csv') #importing the
labels (outputs)
```

Next, the dataset is shuffled and then divided in a training set and in the testing set, which will later measure how well the model was trained.

```
from sklearn.model_selection import
train_test_split
X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.3,
random_state=74)
```

Next, the inputs or features are normalized in a manner that disallows large discrepancies between input magnitudes.

```
from sklearn.preprocessing import
MinMaxScaler
scaler = MinMaxScaler()
scaler.fit(X_train)
```

```
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
```

Here, the Deep Neural Networks are built with 4 layers. The first one contains 88 neurons, and the last one contains 45 neurons. 45 coincides with the number of possible tap positions of the second feeder, which has 9 transformers equipped with E-OTLCs.
Activation functions used here represent the probabilistic functions of that neuron to assume a certain output. If the inputs are low or negative, that neuron may not 'fire' its own output, which at the same time is an input for the next layer.
Sigmoid is similar to Heaviside (Unit step) function, however, it is smoother and continuous around the y-axis.

```
from tensorflow.keras.models import
Sequential
from tensorflow.keras.layers import
Dense, Dropout

model = Sequential()

model.add(Dense(88, activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(75, activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(58, activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(45,
activation='sigmoid'))

model.compile(optimizer='adam',loss='bina
ry_crossentropy')
```

Dropout randomly turns on and off a portion of the neurons to find better results. Here, the model is trained and taught to stop early if the loss function does not improve after 25 epochs.

```
from tensorflow.keras.callbacks import
EarlyStopping
early_stop =
EarlyStopping(monitor='val_loss',
mode='min',verbose=1, patience=25)

model.fit(x=X_train, y=y_train,
epochs=600,
validation_data=(X_test,y_test),callbacks
=[early_stop])
```

Finally, the results of the DNN training are shown below.

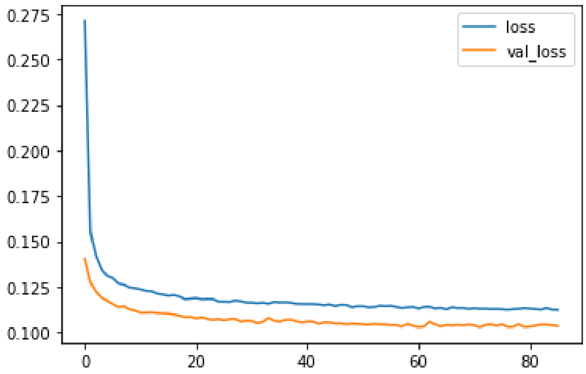

Figure 19. Feeder 2 - Training the Deep Neural Networks

The Deep Neural Network takes inputs in the input layer, then moves along the layers multiplying the inputs to the weights and biases of each neuron, when it reaches the end, it compares the its results to the actual results. According to that comparison, which in mathematical terms can be different, depending on the loss function type, the values move back from the output towards the input, adjusting weights and biases (backpropagation). This cycle is called an Epoch. With each epoch, the weights of neurons are adjusted in such a manner that the predicted outputs tend to be equal to the real ones, as measured by the loss function. In Figure 19, it is notable how the algorithm has a very steep decline of the loss function within the very first epochs. That is reasoned by the fact that DNNs are a robust tool which are able to very quickly 'figure out' the mathematical inter-relations of power system's parameters such as power flows, voltages, loads, etc.

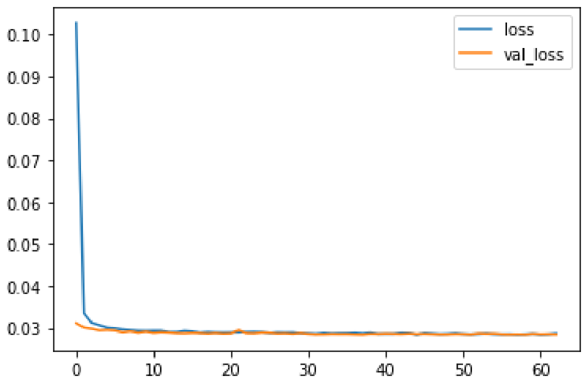

Figure 20. Feeder 1 - Training the Deep Neural Networks.

The model, code, and results of the first studycase are almost identical as those of the second studycase discussed above, hence they were mostly skipped. Different activation functions, number of neurons, rates of Dropout, etc., were tried as attempts to improve the algorithm. The results do not exhibit any significant differences.

A perhaps more useful representation is if instead of loss functions, voltages of a feeder working based on the DNN model are shown graphically. That can be done by using a dataset that the model has not trained on before. In this case, 5000 sets of randomly shuffled inputs that were fed to the model, then the DNN model has predicted the tap positions of each transformer for each of those datasets. These tap positions that are predicted by the DNN model are then used to create characteristics for each transformer equipped with E-OLTC. After that, another Quasi-Dynamic simulation is run, this time the difference is that it only runs around 5000 load flows (hourly data), and more importantly, the tap positions are now inputs together with loads and PVs, instead of outputs as they were in previous Q-D runs. This means, PowerFactory's algorithm of guessing the tap positions is replaced by the DNN's predictions, after which the voltages are monitored and graphed as seen below.

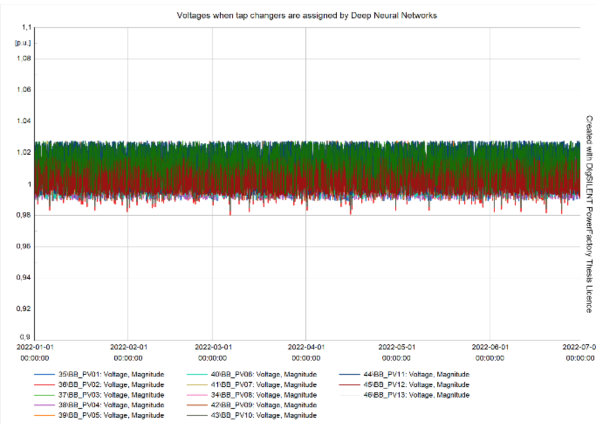

Figure 21. Feeder 1 voltages when the DNN predicts the taps positions [p.u].

Figure 21 confirms what the loss function is showing, which is that the DNN has been able to train so well, it can predict the tap positions in such a manner that allow no voltage constraints to be broken, and it performs as good as PowerFactory's equivalent shown in Figure 13. On the other hand, Figure 22 shows the voltages of the second feeder. They are as good as ones of the first feeder. The differences in range are reasonably justified by the feeder's design and parameters even before the ATPs and DNNs were used. To have a better understanding of the performance of the DNN on guessing the tap positions of the second feeder, Figure 22 should be compared with Figure 15.

Figure 22. Feeder 2 voltages when the DNN predicts the tap positions [p.u].

7. Conclusions

The two studycases carried out in this study reach two conclusions. First one is the fact that E-OLTCs can have immense impact in voltage regulation of MV networks with high penetration of DGs, even without the help of some sort of reactive power control method. The second conclusion is that DNNs can be used to find the correct tap positions of all transformers in a feeder, instead of classical approach of software modelling and running load flows. The results show that DNNs can learn fast and in such a manner that voltage constraint violations are almost eliminated or completely eliminated. The first studycase shows that the loss function was already significantly lower than the loss function of the second studycase, that makes perfect sense provided that the first studycase had a much lower number of voltage constraints violations even before using ATPs.

Although between the two studycases there are around 5.4 million data points, the computational part is quite cheap in terms of memory requirements and time it consumes. The code is also short and flexible, allowing to fill any missing data from measurements or add new features/inputs. The trained algorithm can be saved and programmed to learn constantly from new examples. This makes it possible to regulate the voltage without any human input, provided that the model has access to measurements or at least accurate predictions of loads and PV outputs in the feeder.

DSOs could apply similar methods by installing these trained models in computers of control rooms in HV/MV substation. The model could be programmed to act on its own, thus saving on costs of expensive software licenses and staff training.
The ML algorithm is also programmed in a manner that in the future, it can easily integrate other voltage control methods such as reactive power control ones and combine them with the existing methods of this case to enhance even further its performance.

References

[1] "https://www.iea.org/data-and-statistics/charts/evolution-of-solar-pv-module-cost-by-data-source-1970-2020".

[2] "Directive (EU) 2018/2001 of the European Parliament and of the Council of 11 December 2018 on the Promotion".

[3] V. V. T. R. B. J. N. A. Woyte, "Voltage fluctuations on distribution level introduced by photovoltaic systems," *IEEE Transaction on Energy Conversion v.21, n.1,* 03 2006.

[4] J. R. A. F. Katiraei, "Solar PV Integration Challenges," *IEEE Power and Energy Magazine vol 9, n.3,* 04 2011.

[5] N. H. J. M. P. D. a. N. J. J. Lopes, "Integrating distributed generation into electric..."..

[6] K. Z. a. B. M. I. S. Abapour, Dynamic Planning of Distributed Generation Units in Active Distribution Network, 2015.