



COMILLAS
UNIVERSIDAD PONTIFICIA

ICAI

MASTER'S DEGREE IN INDUSTRIAL ENGINEERING

MASTER THESIS

**ANALYSIS AND PREPARATION OF THE
DEPLOYMENT OF PRIME 1.4 SMART METERS IN
AN INTEROPERABLE FIELD ENVIRONMENT FOR
15,000 METERS**

Autor: Borja Villasante Martín

Director: Alberto Sendín Escalona

Director: Javier Matanza Domingo

Madrid

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

Analysis and preparation of the deployment of PRIME 1.4 smart meters in an interoperable field environment for 15,000 meters

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el curso académico 2020/2021 es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.



Fdo.: Borja Villasante Martín

Fecha: 24/08/2021

Autorizada la entrega del proyecto

LOS DIRECTORES DEL PROYECTO

Fdo.: Alberto Sendín Escalona

Fecha: 24/08/2021



Fdo.: Javier Matanza Domingo

Fecha: 24/08/2021



COMILLAS
UNIVERSIDAD PONTIFICIA

ICAI

MASTER'S DEGREE IN INDUSTRIAL ENGINEERING

MASTER THESIS

**ANALYSIS AND PREPARATION OF THE
DEPLOYMENT OF PRIME 1.4 SMART METERS IN
AN INTEROPERABLE FIELD ENVIRONMENT FOR
15,000 METERS**

Autor: Borja Villasante Martín

Director: Alberto Sendín Escalona

Director: Javier Matanza Domingo

Madrid

Acknowledgments

I would like to thank the following people, who have helped me complete this Master Thesis and/or finish my studies in engineering after six long hardworking years.

To my parents Andoni and Esther and my brother Guillermo, for being the ones that love me the most. We have celebrated the successes together; they have supported me in the hardest moments and their effort has allowed me to get an education and achieve my goals in life. They have always been there for me and I will also be there for them.

To my uncles Luis Mari and Isabel and my cousins Emilio and Fernando, for their hospitality, love, care and support during my two years of stay in Madrid, providing a welcoming environment that allowed me to fully focus on my studies and being there in the good and bad moments.

To my family in general, for always being there for me and becoming the cornerstone that supports me and gives me strength.

To Alberto, Ainara and Javier, for guiding me during this Master Thesis, dedicating their time to supporting and teaching me many new things and making the internship an enjoyable experience.

To my classmates, for the hundreds of hours we have stayed together in the university, the many things I have learned from them and the good moments we have shared together.

To my friends, for being my second family and source of happiness, with whom I want to continue sharing experiences.

ANALYSIS AND PREPARATION OF THE DEPLOYMENT OF PRIME 1.4 SMART METERS IN AN INTEROPERABLE FIELD ENVIRONMENT FOR 15,000 METERS

Autor: Villasante Martín, Borja.

Director: Sendín Escalona, Alberto.

Director: Matanza Domingo, Javier.

Entidad Colaboradora: Iberdrola

RESUMEN DEL PROYECTO

Este proyecto se enmarca en la fase previa al primer despliegue masivo de contadores PRIME v1.4, diseñando la estrategia para estructurarlo. Se han analizado la topología eléctrica y el funcionamiento de la tecnología PLC con PRIME v1.3.6, monitorizando todos los centros de transformación (CTs) y después categorizándolos para estructurar el despliegue en fases con dificultad incremental, asegurando así una buena transición a PRIME v1.4 y maximizando el aprendizaje.

Palabras clave: Centro de transformación, Contadores inteligentes, Estrategia de despliegue, NB-PLC, PRIME v1.4, Red de baja tensión, Smart Grid.

1. Introducción

Los despliegues de contadores inteligentes son considerados como una de las bases para el desarrollo futuro de la red inteligente (Smart Grid).

Entre las tecnologías de telecomunicaciones existentes, la comunicación de banda estrecha por líneas de potencia (NB-PLC) es una de las más populares, dada su buena relación entre rendimiento en campo y coste. PRIME (PowerLine Intelligent Metering Evolution) es el standard ITU-T G.9904 [1], desarrollado por la PRIME Alliance [2] y actualmente instalado en la red de baja tensión de Iberdrola. PRIME conecta el concentrador, situado en el centro de transformación (CT), y los contadores inteligentes, situados en las casas de los clientes, permitiendo una comunicación bidireccional.

Durante la última década, Iberdrola ha desplegado PRIME de manera extensa y exitosa en su red de baja tensión en España. Sin embargo, la necesidad de mejorar las capacidades actuales de PRIME para poder añadir los nuevos servicios y aplicaciones de la Smart Grid ha propiciado el desarrollo de PRIME v1.4. Esta última versión de PRIME incluye muchas mejoras, destacando la extensión de banda 3-95 kHz a 3-500 kHz, mejorando la flexibilidad y permitiendo velocidades superiores, y la adición de modulaciones más robustas para incrementar la fiabilidad en condiciones ruidosas.

A la hora de realizar un despliegue de contadores inteligentes siempre hay mucha incertidumbre debido a la ausencia de una metodología clara aplicable a la mayoría de los casos. Las razones para ello son la dificultad de acceso a la red eléctrica de los académicos, la poca información disponible de manera pública y la heterogeneidad de la red de baja tensión, no existiendo dos redes iguales. Además, al contrario que con otras tecnologías, en un despliegue PLC no pueden situarse los equipos de telecomunicaciones en el lugar óptimo, ya que los contadores siempre deben estar en las casas de los clientes y el concentrador en el CT [5].

Por todo ello, las planificaciones previas son poco útiles en los despliegues de PLC, siendo la incertidumbre máxima, y por ello, la estrategia seguida por Iberdrola en despliegues pasados consiste en realizar unas monitorizaciones inmediatamente posteriores al despliegue y analizar los resultados, resolviendo así los problemas encontrados lo más rápido posible [5]-[6].

2. Definición del proyecto

Este proyecto se enfrenta a la estructuración del primer despliegue masivo de la tecnología PRIME v1.4, mejorando la metodología empleada por Iberdrola en sus despliegues pasados [6] y adaptándola a PRIME v1.4.

La motivación principal del proyecto es el interés de Iberdrola en realizar su primer despliegue con PRIME v1.4 de manera exitosa, garantizando una transición fluida desde PRIME v1.3.6, consiguiendo un buen rendimiento lo antes posible. Además, se cuenta con la ventaja de que PRIME v1.3.6 está actualmente instalado en campo, permitiendo esto analizar el funcionamiento PLC actual en los CTs del despliegue y extraer información valiosa, evitando así planear el despliegue a ciegas.

Tres municipios españoles, a los cuales se va a denominar como “North”, “Central” y “East” (Figura 1), han sido los escogidos para este primer despliegue dadas sus características adecuadas para el primer piloto masivo de PRIME v1.4 en campo.



Figura 1. Municipios del primer despliegue PRIME v1.4.

3. Desarrollo del Proyecto

Las cinco fases del proyecto, ordenadas en orden cronológico, son las siguientes:

A. Estudio de la topología de la red eléctrica

Toda la información necesaria para este estudio está disponible en las bases de datos de Iberdrola, siendo la dificultad encontrarla y deducir cuál es útil realmente.

De entre cientos de variables, se ha hecho una selección de 29, p. ej. la cantidad de clientes, la potencia contratada, las longitudes de línea, etc. Posteriormente, se ha realizado un análisis estadístico para ver las diferencias existentes entre los municipios y las correlaciones entre las variables estudiadas.

Finalmente, se han seleccionado tres de ellas para crear una primera clasificación preliminar de CTs con topología eléctrica similar, y por tanto probablemente también

funcionamiento PLC. Para comprobar la validez de esta primera clasificación, se ha empleado una variable de rendimiento PLC disponible en bases de datos, el éxito medio diario en la lectura de los contadores.

B. Monitorización del funcionamiento PLC

Se ha realizado una monitorización completa de los 159 CTs para entender su funcionamiento PLC actual con PRIME v1.3.6. Esto permite establecer una base de comparación para cuando se instale PRIME v1.4 (el objetivo es mejorar) y también estructurar el despliegue, identificando los CTs más fáciles y difíciles.

El método de monitorización empleado ha sido el desarrollado por Iberdrola para sus despliegues pasados [6], pero las herramientas empleadas requerían ser adaptadas y por ello fueron modificadas y testeadas hasta funcionar correctamente. Monitorizar un único CT es un proceso laborioso que abarca 10 tareas y dura una semana, por lo que automatizarlo es crítico (159 CTs). La monitorización registra lo siguiente:

- **Topología:** Registra la evolución de la topología PRIME, siendo esta los contadores conectados y desconectados en cada instante durante el periodo de monitorización y sus estados funcionales (“switch”, “terminal”, etc.) [4].
- **Ciclos cortos/largos:** Ciclos repetidos de solicitud de lecturas simples (cortos) o múltiples (largos) a los contadores. Se registran tanto el porcentaje de éxito de los ciclos como sus duraciones.

C. Postprocesado y diseño de nuevas variables

La información bruta extraída tras monitorizar son cientos de archivos csv por CT. Para obtener resultados entendibles, es necesario un postprocesado para calcular variables de alto nivel que permitan comprender el funcionamiento PLC del CT.

Este postprocesado ya fue desarrollado anteriormente por Iberdrola, pero nuevamente, su adaptación era necesaria. Además, se ha añadido un procesado adicional empleando Python para calcular unas variables de rendimiento PLC de nuevo diseño, algunas de ellas específicas para PRIME v1.4. En total, incluyendo las nuevas, se han calculado 48 variables PLC. Los aspectos más importantes determinados por las variables son [6]:

- **Disponibilidad:** Referida a la topología (disponibilidad PRIME), mide el porcentaje del tiempo que los contadores están conectados a la red, y referida a los ciclos (disponibilidad de ciclos cortos/largos), mide el porcentaje de éxito de lectura de los contadores promedio de la red.
- **Estabilidad:** Capacidad para mantener un número estable de contadores conectados.

D. Categorización de los centros de transformación

De entre las 48 variables calculadas, se han seleccionado cinco para tratar de agrupar los CTs con funcionamiento similar en categorías, siendo estas las siguientes: contadores conectados (media y desviación típica SD), disponibilidades PRIME (media y SD), disponibilidades de los ciclos cortos/largos (media y SD) y un histograma de estabilidad (Figura 2), una de las variables de nuevo diseño y por tanto a destacar.

Los contadores pueden cambiar de estado funcional y sus conexiones con otros contadores durante el funcionamiento. Este histograma registra el porcentaje de tiempo que cada contador del CT pasa en su estado más común, y luego los separa en rangos de estabilidad (100-90%, 90-80%, ...,10-0%), siendo una variable de 10 valores. Finalmente, se dividen los 10 valores entre el n.º total de contadores, obteniendo así el porcentaje de contadores por rango de estabilidad. Por ejemplo, el CT1 de la Figura 2 es más estable que el CT2 porque tiene más contadores en los rangos más estables (pico en el lado izquierdo). La situación ideal y más estable sería tener el 100% de contadores en el rango 100-90%.

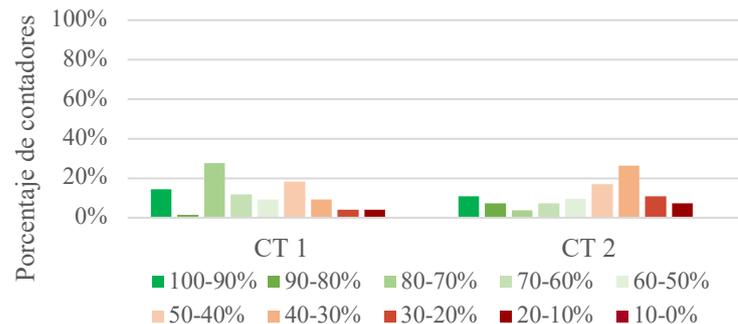


Figura 2. Variable de rendimiento PLC - Histograma de estabilidad.

El agrupamiento de los CTs se ha realizado empleando el método de Machine Learning k-means [7]. Se han probado cinco combinaciones diferentes de las variables de entrada al algoritmo de entre las mencionadas previamente y se ha seleccionado la óptima para realizar el despliegue. También se han probado distintos números de grupos por opción. Adicionalmente, se ha empleado la técnica denominada análisis de componentes principales (PCA) [8] con los resultados del agrupamiento para poder visualizar los grupos en representaciones 2D (Figura 3) o 3D y verificar su validez.

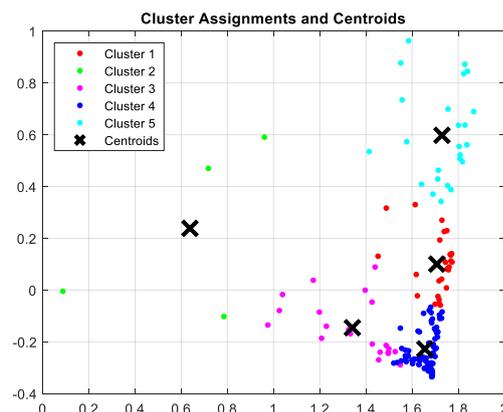


Figura 3. Ejemplo de agrupamiento - Grupos y centroides.

E. Estrategia de despliegue

El despliegue se ha estructurado en distintas fases para asegurar buena transición a PRIME v1.4 y optimizar el aprendizaje, incrementando la dificultad progresivamente. Las decisiones a tomar son las fechas de cada fase, los CTs que la conforman, la marca de contadores a instalar y sus cantidades y el concentrador a instalar. Los factores que

influyen en la estrategia de despliegue son las fechas de recepción de los equipos, las categorías de CT creadas, el tiempo y aspectos específicos de algunos CTs.

4. Resultados

A. Estudio de la topología de la red eléctrica

Los municipios Central y East parecen a priori más favorables, siendo las líneas aéreas más cortas, habiendo más clientes por CT, menos cajas generales de protección por línea y estando los contadores más concentrados. En general, la densidad de población de todos ellos no es elevada, y los consumos principales son viviendas y alguna que otra área industrial.

Además, la presencia de variables muy correlacionadas, p. ej. la potencia contratada y el número de clientes, ha permitido detectar variables redundantes y proponer una clasificación empleando sólo tres de ellas (el número de clientes, los clientes por caja general de protección y la distancia entre cajas generales de protección), las cuales son relevantes para PLC y no están correlacionadas.

Esta primera clasificación no ha resultado ser lo suficientemente buena (Tabla 1), ya que algunas de las categorías formadas contienen CTs con éxito medio diario de lecturas de contador muy dispares (A y B) y otras muestran unos resultados globales demasiado similares (D, E y F) como para ser consideradas categorías independientes. Por ello, la monitorización de los CTs se ha considerado necesaria para obtener información más detallada del funcionamiento PLC y conseguir una mejor clasificación.

Tabla 1. Clasificación preliminar de CTs

		Customers			
		Customers < 25		25 ≤ Customers ≤ 100	Customers > 100
		Length per fuse box ≥ 100	Length per fuse box < 100		
Customers per fuse box	Customers per fuse box < 3	CATEGORY A 20 SSs Average: 89.08% SD: 24.82%	CATEGORY B 41 SSs Average: 97.85% SD: 5.80%	CATEGORY C 33 SSs Average: 97.28% SD: 3.62%	CATEGORY D 15 SSs Average: 98.93% SD: 1.15%
	Customers per fuse box ≥ 3			CATEGORY E 17 SSs Average: 99.47% SD: 0.91%	CATEGORY F 36 SSs Average: 98.6% SD: 2.28%

B. Monitorización del funcionamiento PLC

A partir de los resultados de las monitorizaciones, se ha demostrado el buen funcionamiento de PRIME v1.3.6 en campo (Tabla 2):

Tabla 2. Resultados de disponibilidad media por municipios.

Municipio	Disponibilidad PRIME (%)	Disponibilidad ciclos cortos (%)	Disponibilidad ciclos largos (%)
North	92,2%	84,6%	82,0%
Central	97,0%	94,5%	93,4%
East	92,7%	86,3%	87,0%
Global	94,1%	88,7%	87,5%

En general, se observa un rendimiento superior en el municipio Central, indicando que es un entorno más favorable para PLC. Sin embargo, todos los municipios muestran unos resultados muy respetables. Adicionalmente, se han realizado otras estadísticas agrupando los CTs por fabricante de concentrador, consumo predominante, etc. buscando encontrar tendencias y extraer nuevas conclusiones.

C. Categorización de los centros de transformación

Algunas opciones dan una importancia excesiva a las disponibilidades o a la estabilidad, y otras no consideran el tamaño del CT (contadores conectados), algo importante para planear el despliegue. Finalmente, se ha seleccionado la opción n.º 5, la cual usa conjuntamente las cinco variables mencionadas previamente, obteniéndose la mejor relación entre tamaño, disponibilidad y estabilidad en los grupos formados. Además, el histograma de estabilidad, la única variable de nuevo diseño empleada, ha resultado ser determinante para poder obtener una buena clasificación de CTs, permitiendo medir la estabilidad de manera novedosa y diferenciar mejor los CTs.

La Tabla 3 muestra los resultados completos de los grupos para la opción n.º 5, mostrando los promedios de cada variable por grupo. Además, la Tabla 4 resume la información de la Tabla 3, indicando las características de los CTs que pertenecen a cada grupo, algo crucial para estructurar el despliegue:

Tabla 3. Opción de agrupamiento seleccionada - Resultados promedio y número de CTs por grupo.

Opción 5	Contadores conectados		Disp. PRIME (%)		Disp. ciclos Cortos (%)		Disp. ciclos largos (%)		Histograma de estabilidad										N.º CT
	Media	SD	Media	SD	Media	SD	Media	SD	100-90%	90-80%	80-70%	70-60%	60-50%	50-40%	40-30%	30-20%	20-10%	10-0%	
1	27	6	98,4%	4,2%	98,9%	3,3%	96,1%	9,5%	59,2%	6,3%	9,0%	5,7%	6,4%	6,0%	3,0%	0,9%	0,0%	3,5%	23
2	101	17	98,7%	3,9%	96,9%	7,8%	96,4%	8,9%	27,8%	11,3%	9,7%	12,8%	13,7%	11,1%	7,0%	4,4%	0,8%	1,5%	39
3	25	3	94,9%	9,5%	91,4%	15,6%	88,4%	18,2%	9,6%	9,0%	10,8%	14,5%	13,7%	15,2%	16,2%	7,9%	2,6%	0,3%	36
4	301	51	95,4%	8,7%	88,7%	13,0%	87,4%	14,0%	15,6%	7,2%	10,5%	11,3%	12,3%	15,0%	13,2%	10,3%	4,0%	0,5%	29
5	50	11	78,8%	21,2%	63,5%	30,3%	66,1%	28,6%	5,6%	7,1%	8,0%	10,4%	13,9%	12,2%	14,0%	19,1%	9,4%	0,5%	14
6	91	25	60,8%	9,1%	15,9%	5,1%	20,9%	10,9%	23,4%	6,6%	2,7%	5,0%	5,6%	4,9%	6,4%	20,7%	24,2%	0,5%	4

Tabla 4. Resultados del agrupamiento – Resumen.

Grupo	Tamaño	Disponibilidad	Estabilidad	N.º SS
1	Pequeño	Excelente	Excelente	23
2	Medio-Pequeño	Excelente	Muy buena	39
3	Pequeño	Muy buena	Good	36
4	Grande	Muy buena	Good	29
5	Medio-Pequeño	Mala	Mala	14
6	Cualquiera	Muy mala	Muy mala	4

D. Estrategia de despliegue

Se han estructurado las fases del despliegue con dificultad incremental, instalando los CTs de los grupos con mejores resultados en las primeras fases y acumulando aprendizaje para enfrentarse a los casos más complejos:

1. **Pruebas individuales por fabricante:** Se comprueba que los contadores de cada fabricante funcionan bien por separado, empleando los grupos 1 y 2. Sólo se despliegan tres o cuatro CTs por fabricante.
2. **Primera prueba de interoperabilidad incremental:** Se selecciona un CT del grupo 1 o 2 y se instalan inicialmente contadores de dos fabricantes. Una vez se obtienen buenos resultados se instala el siguiente fabricante.
3. **Interoperabilidad en CTs medianos y pequeños:** Mezclas completas con CTs de los grupos 1, 2, 3, 5 o 6.
4. **Interoperabilidad en CTs grandes:** Mezclas completas con CTs del grupo 4.

Es importante tener éxito antes de avanzar a la siguiente fase, aunque el tiempo puede favorecer el avance, aunque los resultados aún no sean los suficientemente buenos.

5. Conclusiones

Los despliegues de contadores inteligentes siempre se han caracterizado por su incertidumbre, dada la heterogeneidad de la red eléctrica y la dificultad para predecir los resultados hasta que la instalación no se ha llevado a cabo. Por ello, este primer despliegue PRIME v1.4, a pesar de ser el primero de su clase, cuenta con la ventaja de tener PRIME v1.3.6 instalado en campo, permitiendo esto identificar el comportamiento de los CTs y estructurar la estrategia del despliegue de manera óptima.

En cuanto al estudio de la topología eléctrica realizado, la conclusión principal es que es muy complicado obtener una buena clasificación de CTs a partir de únicamente variables eléctricas, ya que hay externalidades no reflejadas en las variables. De todos modos, este tipo de información es imprescindible para organizar el despliegue, ya que, aunque no se emplee directamente para categorizar los CTs, sí es necesaria para seleccionar uno u otro dentro de la misma categoría y elegir el que más convenga.

En cuanto a la monitorización de los centros y el procesado de los datos, sería muy recomendable automatizar más el proceso, ya que aún con las herramientas disponibles sigue siendo bastante laborioso. En cuanto a las nuevas variables de rendimiento PLC introducidas, seguramente serán muy útiles para poder analizar el comportamiento de PRIME v1.4 y realizar la comparación con PRIME v1.3.6, buscando mejorar.

En cuanto a la categorización de los centros, se ha demostrado que pueden obtenerse buenos resultados empleando técnicas de Machine Learning y las variables seleccionadas, obteniendo grupos de CTs con estabilidad, disponibilidad y tamaño muy similares. Además, lo más destacable es el éxito de la única variable de nuevo diseño empleada para el agrupamiento, el histograma de estabilidad. Esta variable ha resultado ser imprescindible para poder categorizar los CTs y se ha convertido en una nueva métrica para medir la estabilidad de los mismos, demostrando además una elevada correlación con la disponibilidad. Por todo ello, Iberdrola ha considerado de elevado interés seguir estudiándola, empleando más datos de monitorizaciones pasadas para analizar si las conclusiones obtenidas en este proyecto se mantienen.

Por último, en cuanto a la estructuración del despliegue, la combinación entre las categorías de CTs previamente diseñadas y las fases organizadas con dificultad

incremental, contando con CTs adecuados ya clasificados para cada una de ellas, sin duda la han facilitado enormemente. Esta estrategia es la culminación del proyecto, estructurando el primer despliegue masivo de la tecnología PRIME v1.4 con la menor incertidumbre posible y dando un paso hacia la Smart Grid.

6. Referencias

- [1] ITU-T, “Narrowband orthogonal frequency division multiplexing power line communication transceivers for PRIME networks”, Recommendation G.9904. Available online: <http://www.itu.int/rec/T-REC-G.9904-201210-I/en>
- [2] PRIME Alliance. Available online: <https://www.prime-alliance.org/>.
- [3] PRIME Alliance, “PowerLine Intelligent Metering Evolution, v1.4.” Available online: https://www.prime-alliance.org/wp-content/uploads/2020/04/PRIME-Spec_v1.4-20141031.pdf.
- [4] PRIME Alliance, “PRIME v1.4 White Paper,” Available online: https://www.prime-alliance.org/wp-content/uploads/2020/04/whitePaperPrimeV1p4_final.pdf.
- [5] A. Sendin, I. Peña, and P. Angueira, “Strategies for Power Line Communications Smart Metering Network Deployment,” *Energies*, vol. 7, no. 4, pp. 2377–2420, Apr. 2014.
- [6] A. Sendin, I. Berganza, A. Arzuaga, A. Pulkkinen and I. H. Kim, "Performance results from 100,000+ PRIME smart meters deployment in Spain," 2012 IEEE Third International Conference on Smart Grid Communications (SmartGridComm), 2012, pp. 145-150, doi: 10.1109/SmartGridComm.2012.6485974.
- [7] P. Sharma, “The Most Comprehensive Guide to K-Means Clustering You’ll Ever Need,” *Analytics Vidhya*. Available online: <https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/>
- [8] J. Lever, M. Krzywinski, N. Altman, “Principal component analysis,” *Nature Methods*, vol. 14, no. 7, pp. 641–642, July 2017. Available online: <https://www.nature.com/articles/nmeth.4346.pdf>

ANALYSIS AND PREPARATION OF THE DEPLOYMENT OF PRIME 1.4 SMART METERS IN AN INTEROPERABLE FIELD ENVIRONMENT FOR 15,000 METERS

Author: Villasante Martín, Borja.

Supervisor: Sendín Escalona, Alberto.

Supervisor: Matanza Domingo, Javier.

Collaborating Entity: Iberdrola

ABSTRACT

This project faces the task of structuring the first massive PRIME v1.4 deployment, designing the strategy to be followed. Analyses of the electric topology and PRIME v1.3.6 performance have been conducted, monitoring all the secondary substations and categorizing them to structure the deployment in stages with incremental difficulty, ensuring a smooth transition and maximizing the knowledge gained for future deployments.

Keywords: Deployment strategy, Low Voltage grid, NB-PLC, PRIME v1.4, Secondary substation, Smart grid, Smart meters.

1. Introduction

Smart metering deployments are considered as one of the foundations for the development of the Smart Grid, enhancing the operation and supervision capabilities of the low voltage (LV) grid.

Among the different existing communication technologies, Narrowband Power Line Communication (NB-PLC) is one of the most popular for smart metering due to its cost effectiveness, having reasonable installation, operation and maintenance costs. PRIME (Powerline Intelligent Metering Evolution) is the ITU-T G.9904 standard [1] developed within the PRIME Alliance [2] and currently deployed in Iberdrola's low voltage (LV) grid. PRIME connects the concentrator, located at the secondary substation (SS), and the smart meters, located at the customers' homes, enabling two-way communications.

During the last decade, Iberdrola has extensively and successfully deployed PRIME v1.3.6 in the Spanish LV grid. Nonetheless, the necessity to improve the current capabilities to support the new SG services and applications has led to the development of PRIME v1.4 [3]. The latest version of the standard includes many improvements, being the increase of bandwidth from 3-95 kHz to 3-500 kHz, providing flexibility and increased throughput, and the addition of new robust modes to increase reliability the most relevant ones [4].

When facing a smart meter deployment, even if they are not unusual at all, a lot of uncertainty is faced due to the lack of literature and guidelines applicable to all or most cases. The reasons for this reality are mainly related to the difficult access to real grids by academics, the lack of data available to the public and the heterogeneity of the electric power grid, two equal grids not existing. Besides, on contrary to any other telecommunication technology deployment, the location of the equipment is already predetermined, as the smart meters must be located at the customers premises and the concentrators at the SSs, not being the optimal places [5].

Consequently, ex-ante planning efforts are almost worthless for PLC deployments, and thus the strategy traditionally followed by Iberdrola for its deployments is a post-deployment monitoring of the PLC performance and analysis of the obtained results to solve the issues present [5]-[6].

2. Project definition

This project faces the task of structuring Iberdrola's first massive PRIME v1.4 deployment, improving the previously developed methodology for preceding deployments [6] and adapting it to the newest version of PRIME.

The main motivation behind is the interest of the utility to perform its first massive PRIME v1.4 successfully, guaranteeing a smooth transition from PRIME v1.3.6, achieving improved performance as soon as possible. Besides, the advantage of having PRIME v1.3.6 already deployed on the field, allowing the analysis of the current performance of the SSs and not planning the deployment blindly is present.

Three Spanish municipalities, which will be referred as North, Central and East (Figure 1), have been selected for the deployment due to their appropriate characteristics for this first on the field "test bed".



Figure 1. First PRIME v1.4 deployment municipalities.

3. Project development

Five different project stages can be distinguished, presented in chronological order:

A. Grid topology study

A thorough research of the electric topology of the SSs was performed. The data is available in the utility's databases and the difficulty lies in finding the useful one.

From hundreds of variables, a selection of 29 was done, i.e. the number of customers, the contracted power, the feeder length, etc. Afterwards, a statistical analysis was done to see the differences between municipalities and the correlation between variables was also studied.

Finally, three variables were selected to create a first draft classification of the SSs with similar topological characteristics and thus probably similar PLC behavior. A simple PLC performance variable that can be directly downloaded from the databases, the

average daily meter reading success, was used to check the validity of the groups formed with this first classification.

B. Power Line Communications performance monitoring

A complete monitoring of the 159 SSs was performed to understand their current PRIME v1.3.6 performance. The results are essential to serve as a benchmark when PRIME v1.4 is deployed, comparing the performance results, and to structure the deployment stages, identifying the easiest and most difficult SSs.

The monitoring methodology had been previously developed by Iberdrola [6], but the required tools were outdated, and they had to be adapted and tested until they functioned correctly. Monitoring a single SSs is a tedious process which consists of 10 tasks and takes a whole week to be completed, so task automation is compulsory. During monitoring, these aspects are registered:

- **Topology:** It registers how the PRIME topology evolves for one week. Connected and disconnected nodes and their functional states (switch, terminal, etc.) [4] are recorded.
- **Short/Long polling cycles:** Repeated cycles to ask all the meters for a simple (short) or multiple (long) reading. Both the success rate and duration of each cycle are registered.

C. Post-processing and new variable design

The raw data obtained after the monitoring consists of hundreds of csv files per SS. To produce humanly understandable results, data post-processing is needed to calculate high level variables that show the PLC performance of the SS.

The post-processing of the data had been previously developed by Iberdrola, but once again, an adaptation was needed. Besides, an additional posterior processing step using Python was also added to calculate some newly developed PLC performance variables. Accounting the new variables, a total of 48 variables were calculated. The most relevant PLC performance metrics to consider are the following [6]:

- **Availability:** If referred to the topology (PRIME availability), it measures the percentage of time that the smart meters are connected to the subnetwork, and if referred to the cycles (short/long cycles availability), it measures the average meter reading success of the subnetwork.
- **Stability:** Ability to maintain a constant number of nodes connected, and if possible, with the same functional states.

D. Secondary substation categorizing

From the 48 PLC performance variables, five were selected to group the SSs into categories with similar PLC performance: connected nodes (avg. and SD), PRIME availability (avg. and SD) short/long cycles availability (avg. and SD) and a stability histogram (Fig. 2), a promising newly designed variable.

During operation, meters can change their functional states and topological dependencies. This histogram registers the percentage of time each meter of the SS stays

in its most common state, and then separates them into stability ranges (100-90%, 90-80%, ..., 10-0%), creating a ten-value variable. Finally, the data is normalized by dividing it by the total number of meters of the SS to be able to compare SSs of different sizes, obtaining the percentage of meters present on each range. For instance, SS1 of Figure 2 is more stable than SS2 because it has more meters in the most stable ranges (peak on the left side). The most stable and thus desirable situation would be to have all the meters in the 100-90% range.

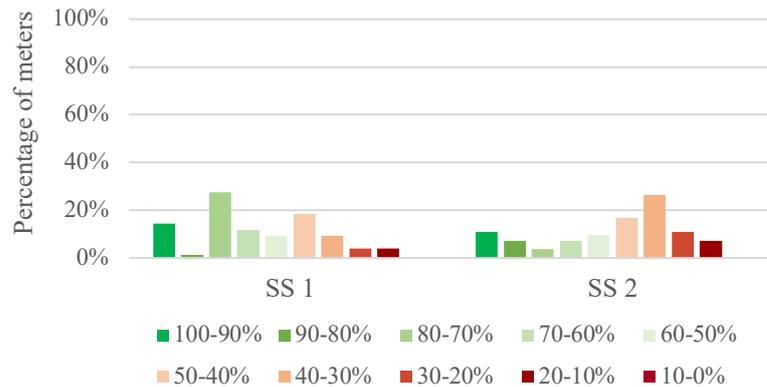


Figure 2. PLC performance variable - Stability histogram.

The SS categorizing was performed using the k-means unsupervised clustering Machine Learning technique [7]. Five different options, consisting on different combinations of the input variables, were tried and the most adequate for deployment purposes was chosen. Different cluster size alternatives were also tried for each option, selecting the one that produced the best results. Additionally, Principal Component Analysis (PCA) [8] was applied to the clustering results to be able to visualize the clusters in 2D (Figure 3) or 3D plots.

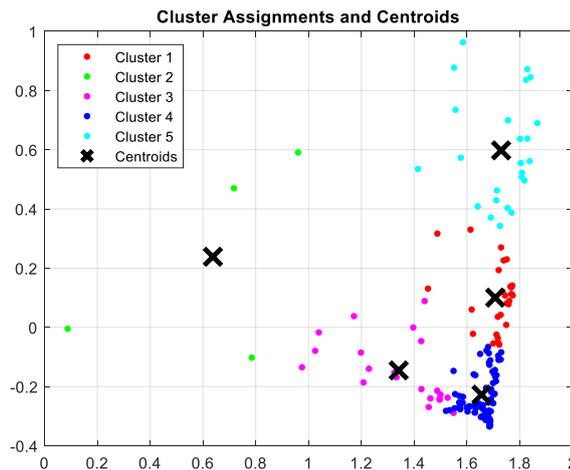


Figure 3. Clustering results example (2D) - Cluster assignments and centroids.

E. Deployment strategy

The deployment was structured in different stages to ensure a smooth transition and optimize the knowledge gained, increasing the complexity in an incremental way. The

specific deployment dates for each stage, the chosen SSs and the brand/s of smart meters and concentrator to install on each SS are the aspects to be decided. The factors affecting the deployment are the equipment arrival dates for each manufacturer, the designed SS categories, time constraints and specific aspects of each SS.

4. Results

A. Grid topology study

Central and East municipalities look more favorable for PLC, having shorter overhead feeder lengths, more customers, less fuse boxes per feeder and more concentrated meters. In general, the three municipalities are not densely populated, and the main loads are customer homes with a few industrial areas also present.

Moreover, the presence of highly correlated variables, i.e. the contracted power, the number of customers and the number of fuse boxes, allowed for the detection of redundant variables and the proposal of a draft classification using only three of them, which are relevant for PLC and not correlated.

The results of this first classification (Table 1) were not good enough, as some categories (A and B) contained SSs with a very different average daily meter reading success (high SD) and others showed very similar global results (D, E and F) and could not be distinguished. For that reason, the monitoring of the SSs is necessary to obtain more detailed information about PLC performance and achieve a better classification.

Table 1. SS draft classification

		Customers			
		Customers < 25		25 ≤ Customers ≤ 100	Customers > 100
		Length per fuse box ≥ 100	Length per fuse box < 100		
Customers per fuse box	Customers per fuse box < 3	CATEGORY A 20 SSs Average: 89.08% SD: 24.82%	CATEGORY B 41 SSs Average: 97.85% SD: 5.80%	CATEGORY C 33 SSs Average: 97.28% SD: 3.62%	CATEGORY D 15 SSs Average: 98.93% SD: 1.15%
	Customers per fuse box ≥ 3			CATEGORY E 17 SSs Average: 99.47% SD: 0.91%	CATEGORY F 36 SSs Average: 98.6% SD: 2.28%

B. Power Line Communications performance monitoring

The monitoring results proved the performance of PRIME v1.3.6 on the field, showing the following figures (Table 2):

Table 2. Average availability results by municipality.

Municipality	PRIME avail. (%)	Short cycle avail. (%)	Long cycle avail. (%)
North	92.2%	84.6%	82.0%
Central	97.0%	94.5%	93.4%
East	92.7%	86.3%	87.0%
Global	94.1%	88.7%	87.5%

In general, superior performance was observed in the Central municipality, indicating that it is more favorable for PLC. Be that as it may, all of them show very respectable results. Additionally, the results were also segregated by concentrator manufacturer, predominant consumption, etc. trying to extract additional conclusions.

C. Secondary substation categorizing

Some of the options tried gave excessive importance to the availabilities or the stability to form the groups, while others did not consider the SS size (connected nodes), something essential for deployment purposes. In the end, option 5, which used all the five variables as inputs, was the best trade-off between size, stability and availability. Besides, the stability histogram, the only new variable used in the clustering, was determining to achieve a good classification, becoming a new reference to measure stability and the key to distinguish the SSs better and form the clusters.

Table 3 shows the complete clustering results for option 5, showing the average values of all the variables for the SSs belonging to each cluster. Furthermore, Table 4 summarizes the information of Table 3, indicating the characteristics of the SSs belonging to each of the six clusters, something crucial to plan the deployment:

Table 3. Selected clustering option – Average results and number of secondary substations per cluster.

Option 5 Cluster number	Connected Nodes		PRIME avail (%)		Short cycle avail (%)		Long cycle avail (%)		Stability histogram										N° SS
	Avg.	SD	Avg.	SD	Avg.	SD	Avg.	SD	100-90%	90-80%	80-70%	70-60%	60-50%	50-40%	40-30%	30-20%	20-10%	10-0%	
1	27	6	98.4%	4.2%	98.9%	3.3%	96.1%	9.5%	59.2%	6.3%	9.0%	5.7%	6.4%	6.0%	3.0%	0.9%	0.0%	3.5%	23
2	101	17	98.7%	3.9%	96.9%	7.8%	96.4%	8.9%	27.8%	11.3%	9.7%	12.8%	13.7%	11.1%	7.0%	4.4%	0.8%	1.5%	39
3	25	3	94.9%	9.5%	91.4%	15.6%	88.4%	18.2%	9.6%	9.0%	10.8%	14.5%	13.7%	15.2%	16.2%	7.9%	2.6%	0.3%	36
4	301	51	95.4%	8.7%	88.7%	13.0%	87.4%	14.0%	15.6%	7.2%	10.5%	11.3%	12.3%	15.0%	13.2%	10.3%	4.0%	0.5%	29
5	50	11	78.8%	21.2%	63.5%	30.3%	66.1%	28.6%	5.6%	7.1%	8.0%	10.4%	13.9%	12.2%	14.0%	19.1%	9.4%	0.5%	14
6	91	25	60.8%	9.1%	15.9%	5.1%	20.9%	10.9%	23.4%	6.6%	2.7%	5.0%	5.6%	4.9%	6.4%	20.7%	24.2%	0.5%	4

Table 4. Clustering results – Summary.

Cluster	SS Size	Availability	Stability	N° SS
1	Small	Excellent	Excellent	23
2	Medium - small	Excellent	Very good	39
3	Small	Very good	Good	36
4	Big	Very good	Good	29
5	Medium - small	Bad	Bad	14
6	Any	Very bad	Very bad	4

D. Deployment strategy

The deployment was structured to be performed with incremental difficulty, selecting the clusters with best performance for the initial stages, guaranteeing that more knowledge is accumulated before facing the worst SSs:

- 1. Individual manufacturer trials:** They check that the smart meters of each manufacturer work well independently, and SSs from clusters 1 or 2 are used. Only three or four SSs are deployed per manufacturer.
- 2. First incremental interoperability trial:** A single SSs from clusters 1 or 2 is selected and meters from two different manufacturers are installed. When decent performance is achieved, a new manufacturer is added incrementally.
- 3. Small and medium SS size interoperability trials:** Full manufacturer mixes on SSs of clusters 1, 2, 3, 5 or 6.
- 4. Big SS size interoperability trials:** Full manufacturer mixes on SSs of cluster 4.

It is important to succeed before continuing with the next stage, but the contrary could also happen if the project needs to continue advancing due to time constraints.

5. Conclusions

Smart meters deployments have always been characterized by their uncertainty, due to the heterogeneity of the LV grid and the difficulty to predict the results until the deployment has been performed. For that reason, this first massive PRIME v1.4 deployment, even if it is the first of its kind, relies on the advantage that PRIME v1.3.6 is already deployed on the field, allowing to identify the current performance on the field and structure the deployment optimally.

With respect to the electric grid study, the main conclusion is that it is very difficult to reach a good enough SS classification by only using electric variables, as there are externalities that are not reflected in the data. Anyways, this information is essential to structure the deployment, as even if it is not used to categorize the SSs, it is needed to select between SSs belonging to the same category and identify the most convenient ones for each deployment stage.

With respect to the monitoring of the SSs and the data post-processing, it would be recommendable to achieve a higher degree of automation, as the process is still tedious.

With respect to the newly designed PLC performance variables, they will surely be useful to analyse the performance of PRIME v1.4 and compare it with PRIME v1.3.6, ensuring that an improvement is achieved with the latest version of the PLC standard.

With respect to the SS categorizing, the possibility of classifying the SSs using Machine Learning techniques with the chosen variables, achieving groups with homogeneous stability, availability and size, which are appropriate to structure the deployment has been proven. Furthermore, the most remarkable aspect is the success of the only newly designed variable used for this end, the stability histogram. This variable has been crucial to categorize the SSs and has become a new reference to measure PRIME stability, also showing high correlation with the availability. Consequently, Iberdrola has shown its interest to continue studying this variable, using past monitoring data from hundreds of SSs to analyse if the conclusions extracted from this project can be generalized.

Finally, with respect to the structuring of the deployment, the deployment stages arranged with incremental difficulty together with the previously designed SS categories, containing SAs with adequate characteristics for each stage, has facilitated the task enormously. This strategy is in fact the culmination of the project, structuring the first massive PRIME v1.4 pilot with the least uncertainty possible and becoming an initial step towards the Smart Grid.

6. References

- [1] ITU-T, “Narrowband orthogonal frequency division multiplexing power line communication transceivers for PRIME networks”, Recommendation G.9904. Available online: <http://www.itu.int/rec/T-REC-G.9904-201210-I/en>
- [2] PRIME Alliance. Available online: <https://www.prime-alliance.org/>.
- [3] PRIME Alliance, “PowerLine Intelligent Metering Evolution, v1.4.” Available online: https://www.prime-alliance.org/wp-content/uploads/2020/04/PRIME-Spec_v1.4-20141031.pdf.
- [4] PRIME Alliance, “PRIME v1.4 White Paper,” Available online: https://www.prime-alliance.org/wp-content/uploads/2020/04/whitePaperPrimeV1p4_final.pdf.
- [5] A. Sendin, I. Peña, and P. Angueira, “Strategies for Power Line Communications Smart Metering Network Deployment,” *Energies*, vol. 7, no. 4, pp. 2377–2420, Apr. 2014.
- [6] A. Sendin, I. Berganza, A. Arzuaga, A. Pulkkinen and I. H. Kim, "Performance results from 100,000+ PRIME smart meters deployment in Spain," 2012 IEEE Third International Conference on Smart Grid Communications (SmartGridComm), 2012, pp. 145-150, doi: 10.1109/SmartGridComm.2012.6485974.
- [7] P. Sharma, “The Most Comprehensive Guide to K-Means Clustering You’ll Ever Need,” *Analytics Vidhya*. Available online: <https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/>
- [8] J. Lever, M. Krzywinski, N. Altman, “Principal component analysis,” *Nature Methods*, vol. 14, no. 7, pp. 641–642, July 2017. Available online: <https://www.nature.com/articles/nmeth.4346.pdf>

Index

Chapter 1. Introduction.....	10
Chapter 2. Technology Description	12
2.1 Power Line Communications (PLC).....	12
2.1.1 Narrowband-PLC (NB-PLC).....	14
2.2 PRIME Standard	15
2.2.1 Physical Layer.....	16
2.2.2 MAC Layer.....	17
2.2.3 PRIME v1.4.....	19
2.3 SS Categorization Methods.....	23
2.3.1 Principal Component Analysis (PCA)	23
2.3.2 K-means	24
Chapter 3. State of the Art.....	26
3.1 NB-PLC Deployment Strategies.....	26
3.2 NB-PLC Performance.....	30
3.2.1 Impedance	30
3.2.2 Noise.....	32
3.2.3 Channel selection.....	35
Chapter 4. Project Definition.....	36
4.1 Motivation.....	36
4.2 Objectives	37
4.3 Planning and Economic Estimate	37
4.3.1 Planning.....	37
4.3.2 Economic Estimate.....	38
Chapter 5. Project Development	40
5.1 Introduction.....	40
5.2 Grid topology study	43
5.2.1 Introduction.....	43
5.2.2 Electric Variable Analysis	46

5.2.3	<i>Draft SS Categorization</i>	49
5.3	PLC Performance Monitoring.....	50
5.4	Post-processing and New Variable Design.....	53
5.4.1	<i>Post-processing</i>	53
5.4.2	<i>New Variable Design</i>	53
5.5	SS Categorizing	56
5.5.1	<i>Clustering Variables</i>	57
5.5.2	<i>K-means Clustering</i>	63
5.6	Deployment Strategy	66
5.7	PRIME v1.4 Monitoring Toolkit Preparation	68
Chapter 6. Results Discussion		70
6.1	Grid Topology Study	70
6.1.1	<i>Electric Topology Statistics</i>	70
6.1.2	<i>Correlation Analysis</i>	76
6.1.3	<i>SS Draft Classification</i>	77
6.2	PLC Performance Monitoring.....	80
6.3	SS Categorizing	81
6.3.1	<i>Clustering Process</i>	81
6.3.2	<i>Results</i>	84
6.4	Deployment strategy	85
Chapter 7. Conclusions and Future Work		89
Chapter 8. Bibliography		91
Annex I: Sustainable Development Goals		95
Annex II: Detailed SS Monitoring		96
Annex III: Monitoring Data Post-Processing		101
Annex IV: Monitoring Results		106
Annex V: Detailed SS Categorizing		109

Annex VI: Scripts **115**

Figure Index

Figure 1. PLC technologies in the electric power grid.....	13
Figure 2. PLC infrastructure for the Smart Grid [9].	14
Figure 3. PLC in the LV power grid.	15
Figure 4. PRIME DPSK modulation types [12].....	17
Figure 5. PRIME subnetwork topology (4 levels).	18
Figure 6. Service Node states and transitions [6].....	19
Figure 7. Type A PHY frame [13]	20
Figure 8. Type B PHY Frame [13].....	20
Figure 9. Block scheme of the PRIME v1.4 robust modes FEC mechanism [13].....	21
Figure 10. PRIME v1.4 frequency bands [13].	21
Figure 11. BC PHY frame [13].	22
Figure 12. K-means clusters example [14].....	24
Figure 13. K-means - Optimal cluster number [15].	25
Figure 14. Attenuation in a rural feeder in the 40-500 kHz frequency range [18].....	31
Figure 15. Impedance measurements on different points of the three phases of an urban feeder [20].	32
Figure 16. Impedance measurements on different points of the three phases of a rural feeder [20].	32
Figure 17. Non-intentional emissions generated by PV inverter vs. emission limits [24]..	34
Figure 18. Non-intentional emissions generated by a wind turbine vs emission limits [24].	34
Figure 19. Project schedule.	38
Figure 20. PRIME v1.4 deployment municipalities.....	41
Figure 21. Project data types.	43
Figure 22. LV grid segment topology.	44
Figure 23. Smart meter types.	46
Figure 24. SS LV grid visualization – GENESIS.	46

Figure 25. Electric topology correlation matrix - 8x8 excerpt.....	49
Figure 26. Typical SS monitoring chronogram.....	51
Figure 27. SS monitoring plan.	52
Figure 28. Connected nodes and switches (A.1.1) - Stable network.	57
Figure 29. Connected nodes and switches (A.1.1) - Periodic noise.....	58
Figure 30. Connected nodes and switches (A.1.1) - Random patterned noise.....	58
Figure 31. Stability histogram (C.1.9) - 4 SS comparison.	60
Figure 32. Stability histogram (C.1.9) – Unstable SSs	61
Figure 33. Stability histogram (C.1.9) – Very stable SSs.	62
Figure 34. Stability histogram (C1.9) – SSs with average stability.	62
Figure 35. Cluster number selection – Option 3.	64
Figure 36. Cluster assignments and centroids in 2D - Option 3 (k=5).	65
Figure 37. Cluster assignments and centroids in 3D - Option 3 (k=5).	65
Figure 38. Electrical topology analysis - N_customers.....	70
Figure 39. Electrical topology analysis - %_trif_customers.	71
Figure 40. Electrical topology analysis – N_customers per type.....	71
Figure 41. Electrical topology analysis - P_total.	72
Figure 42. Electrical topology analysis – Feeder lengths.....	73
Figure 43. Electric topology analysis - N_feeders.	73
Figure 44. Electric topology analysis - N_fusebox_per_feeder.....	74
Figure 45. Electrical topology analysis – N_customers_per_fusebox.	74
Figure 46. Electric topology analysis - Length_per_fusebox.	75
Figure 47. Electrical topology analysis - Load types.....	76
Figure 48. Cluster stability curves – Option 1 (k=4).	82
Figure 49. SS monitoring tasks.	96
Figure 50. Smartssive application with example database.....	97
Figure 51. Cycles FTP directory preparation example.	98
Figure 52. Concentrator tasks deactivation example.	98
Figure 53. Topology activation example.....	99
Figure 54. Short cycle xml file example.	99

Figure 55. Cycles xml upload example.....	99
Figure 56. Long cycle xml file example.	100
Figure 57. Powerful tool Excel file example - Summary sheet.	102
Figure 58. Powerful tool Excel file example - Hierarchy sheet.	102
Figure 59. Powerful tool Excel file example - Connected_Nodes_Graph sheet.....	103
Figure 60. Powerful tool Excel file example - Extra_Information sheet.	104
Figure 61. Post-processing block diagram.	105

Table Index

Table 1. Iberdrola Castellón deployment - SS categories [16].	28
Table 2. Variables for PLC smart metering assessment [2].	29
Table 3. Project economic estimate.	39
Table 4. Number of SSs per municipality.	42
Table 5. Number of concentrator and smart meter manufacturers per municipality.	42
Table 6. Electric topology analysis variables.	47
Table 7. PLC performance analysis variables - New variables (blue).	54
Table 8. Cluster results table example - Option 3 (k = 5).	66
Table 9. PRIME v1.4 equipment arrival.	68
Table 10. Correlated variables.	76
Table 11. SS draft classification.	78
Table 12. Monitoring results by municipality (Summary).	80
Table 13. Clustering option comparison.	83
Table 14. Cluster characteristics - Selected option.	85
Table 15. Detailed deployment stages.	87
Table 16. Sustainable Development Goals.	95
Table 17. Monitoring results by concentrator manufacturer (Summary).	106
Table 18. Monitoring results by municipality and manufacturer (Summary).	107
Table 19. Monitoring results by load type (Summary).	108
Table 20. SS categorizing average results per cluster - Option 1 (k=4).	109
Table 21. SS categorizing average results per cluster- Option 2 (k=5).	110
Table 22. SS categorizing average results per cluster - Option 3 (k=6).	111
Table 23. SS categorizing average results per cluster - Option 4 (k=5).	112
Table 24. SS categorizing average results per cluster - Option 5 (k=6).	113

List of Acronyms

- AMI: Advanced Metering Infrastructure
- ARIB: Association of Radio Industries and Businesses
- BB-PLC: Broadband Power Line Communications
- BC: Backward Compatibility
- BN: Base Node
- CFP: Contention Free Period
- CSMA-CA: Carrier Sense Multiple Access with Collision Avoidance
- D8PSK: Differential 8-Phase Shift Keying
- DBPSK: Differential Binary Phase Shift Keying
- DER: Distributed Energy Resources
- DQPSK: Differential Quadrature Phase Shift Keying
- FCC: Federal Communications Commission
- FTP: File Transfer Protocol
- GIS: Geographic Information System
- IP: Internet Protocol
- ITU: International Telecommunication Union
- LV: Low Voltage
- MAC: Media Access Control
- MV: Medium Voltage
- NB-PLC: Narrowband Power Line Communications
- NIE: Non-Intentional Emissions
- OFDM: Orthogonal Frequency Division Multiplexing
- PCA: Principal Component Analysis
- PHY: Physical
- PLC: Power Line Communications
- PRIME: PowerLine Intelligent Metering Evolution
- PV: Photovoltaic

RM: Robustness Management

SD: Standard Deviation

SDG: Sustainable Development Goals

SG: Smart Grid

SN: Service Node

SNR: Signal to Noise Ratio

SQL: Structured Query Language

SS: Secondary Substation

UNB-PLC: Ultra Narrowband Power Line Communications

Chapter 1. INTRODUCTION

Smart metering deployments are considered as one of the foundations for the development of the Smart Grid (SG). Smart metering enables two-way communications between the meters located at the customer premises and the utility. Apart from the basic ability to remotely read and operate the smart meters, it will also facilitate the addition of new SG services related with energy efficiency such as distribution automation, asset management, demand response and intelligent network agents [1] [2].

Among the different available communication technologies, Power Line Communication (PLC) is one of the preferred technologies for smart metering due to its reasonable procurement, installation and operation/maintenance cost. PLC technologies can be grouped into three categories [3]: Ultra Narrowband PLC (UNB-PLC), operating in the 0.3-3 kHz band; Narrowband PLC (NB-PLC), operating in the 3-500 kHz band and Broadband PLC (BB-PLC), operating in the 1.8-250 MHz band. NB-PLC has proven to be the most suitable technology for smart metering communication purposes in SG environments for many utilities. PRIME (PoweRline Intelligent Metering Evolution) is the ITU-T G.9904 standard developed by the PRIME Alliance [4].

During the last decade, Iberdrola has extensively and successfully deployed PRIME in the Spanish grid. The version of the PRIME protocol that is currently deployed on the field is PRIME v1.3.6 [5], which operates in the CENELEC A band, comprising frequencies between 3 kHz and 95 kHz, and has already proven its performance on the field. Nonetheless, the needs for an increase in bandwidth to support new SG services and avoidance of interferences coming from other devices, the inclusion of new “robust modes” to improve reliability and the possibility to achieve higher throughputs have led to the development of a new version of the standard, PRIME v1.4 [6].

The new possibilities that PRIME v1.4. brings forth, together with some preceding field studies [7], have led Iberdrola to the decision of executing the refresh of its deployment with

the newest version of the standard, when smart meters reach the end of their useful life. This project covers the phase previous to this first deployment, which will require a thorough analysis to ensure that the transition from PRIME v1.3.6 to PRIME v1.4 is as smooth as possible, guaranteeing the best performance in interoperable environments, from the very beginning, which is essential for the utility. The analysis will include the appropriate parametrization of the technology to allow an interoperable multi-vendor roll out.

Chapter 2. TECHNOLOGY DESCRIPTION

This chapter introduces the basics of the main technologies that are part of the project, so that the following chapters can be more easily covered. It includes both PLC technologies, and other auxiliary non PLC-related methods used in the analysis performed in this project.

2.1 POWER LINE COMMUNICATIONS (PLC)

Electric power grid can be used as a medium to carry information from one point to another if they are interconnected by an electric cable. This is exactly the basic concept behind Power Line Communications (PLC) technologies.

Power Line Communications have been used by utilities in electricity grid for over a century. During that period, PLC technologies have evolved continuously from the first generation (ripple control), to the second generation (two-way communications) and finally to the third generation (high data rate OFDM) [8]. Moreover, the recent success of PLC has been due to the cost effectiveness of the technology, mainly because the medium (the electric cables) are already installed.

Nowadays, PLC is used both in MV and LV, with different frequencies being more appropriate depending on the case. When it comes to MV, Broadband (BB) PLC is used, operating in the 1.8-250 MHz band and providing up to 200 Mb/s. When it comes to LV, Narrowband (NB) PLC is the preferred technology, operating in the 3-500 kHz bands and providing up to 1 Mb/s [3].

This project only covers smart metering, which is done in the LV section of the grid and thus using NB-PLC, so BB-PLC will not be further explained. Be that as it may, Figure 1 helps with the explanation of the PLC technologies present on each part of the grid, in an end-to-end PLC based architecture:

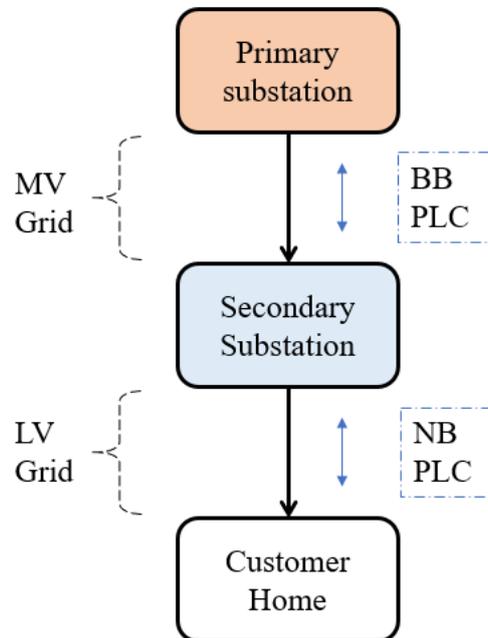


Figure 1. PLC technologies in the electric power grid.

Basically, the NB-PLC subnetwork provides communication between the smart meters, located at the customer homes, and the data concentrator, which is located at the SS. Afterwards, the data from different groups of SS (cells) is concentrated into a single SS using BB-PLC, and finally sent to the central system of the utility. Another alternative option is to communicate the SS with the central system using traditional radio communications.

Moreover, as Figure 1 implies, LV and MV PLC work independently and are joint together at the SS, where the electrical transformer is located. In fact, the electrical transformer is the device that separates LV and MV PLC, as PLC signals have problems to propagate through the electrical transformers and that is also one of the main reasons for this separation.

Finally, Figure 2 provides a good overview of the AMI of the Spanish utility Iberdrola:

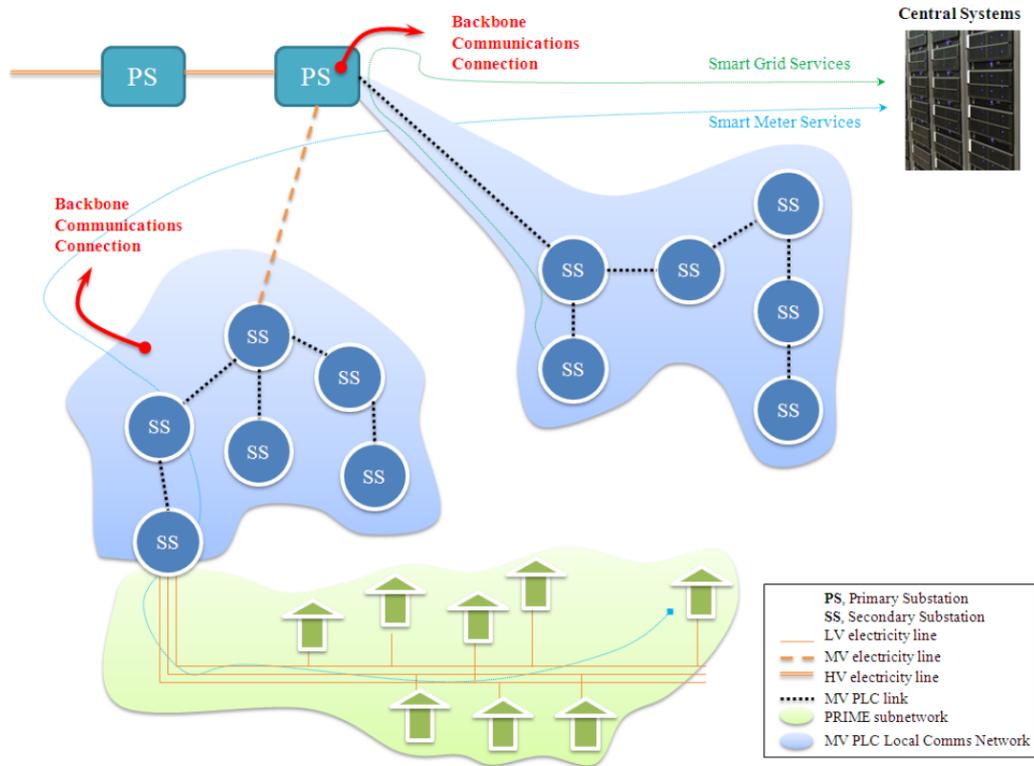


Figure 2. PLC infrastructure for the Smart Grid [9].

2.1.1 NARROWBAND-PLC (NB-PLC)

Among the existing PLC technologies, Narrowband-PLC (NB-PLC) is the one used in the LV part of the grid, and thus the one applicable to smart metering.

A description of a NB-PLC network in a LV grid can be observed in Figure 3.

NB-PLC offers less throughput than BB-PLC since it works in lower frequencies and has less bandwidth available. Be that as it may, it has been proven that NB-PLC has enough capabilities to support smart metering applications and the future SG services and applications.

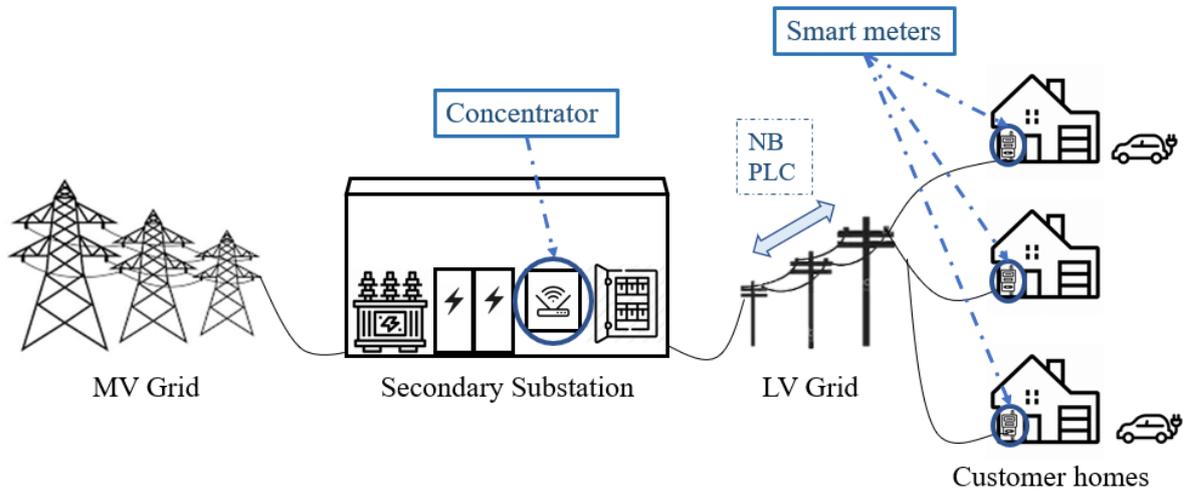


Figure 3. PLC in the LV power grid.

Nowadays, different NB-PLC standards exist in the market, being PRIME (ITU-T G.9904) and G3-PLC (ITU-T G.9903) [10] the most relevant ones. Reference [3] contains a good comparison of four NB-PLC standards, including PRIME and G3-PLC. The evolution of these standards clearly goes towards achieving faster data rates with new modulation types or gaining robustness to combat the noises intrinsic to the electric power grid medium.

2.2 PRIME STANDARD

PRIME (PowerLine Intelligent Metering Evolution) is the focus of this project, as it is the standard used by Iberdrola for smart metering purposes, being also Iberdrola a member of the PRIME Alliance [4].

PRIME was developed within the PRIME Alliance and is now an international standard ITU-T G.9904. Be that as it may, standardization was not a priority before in PLC systems and the focus was to create systems that could perform well in real conditions, which were later promoted to standards due to their adoption or the influence exerted by technology promoters [2].

Furthermore, it is worth mentioning that interoperability is one of the focal points of PRIME, as smart meters and concentrators from different manufacturers can coexist in the same network and communicate without any problem.

The version of the standard that is currently deployed on the field is PRIME v1.3.6 [5], having proven that it can perform successfully. However, a new version of the standard, PRIME v1.4 [6] has been released, which includes new upgrades and is being tested before its first big scale field deployments.

2.2.1 PHYSICAL LAYER

Even if this project doesn't focus on the analysis of variables related to the PHY layer, just the most basic concepts should be kept in mind for a full understanding of the project.

PRIME v1.3.6 uses OFDM modulation in the frequency band from 3 kHz to 95 kHz named CENELEC A-band, reserved for PLC communications compliant with EN 50065-1 [11] in Europe. PRIME v1.4 extends this band up to 500 kHz, covering the FCC and ARIB bands. Be that as it may, frequencies below 40 kHz are always avoided as they are problematic for PLC due to different reasons.

The signal is loaded into 97 equally spaced OFDM subcarriers (96 for data and 1 pilot), which are then transmitted into symbols that last 2240 ms. In the case of PRIME v1.4, as there is more than one channel, adjacent channels are spaced by guard intervals of fifteen subcarriers.

One of the main advantages of PRIME is that different types of modulations can be chosen for the OFDM symbols. Three differential phase shift keying modulation types are possible: DBPSK, DQPSK and D8PSK (Figure 4). DBPSK is the slowest but the most robust of them, while D8PSK is the fastest but least robust, being DQPSK an intermediate option. Their usefulness resides in the fact that depending on channel conditions the most optimal one can be chosen by the Robustness management (RM) algorithm of the protocol.

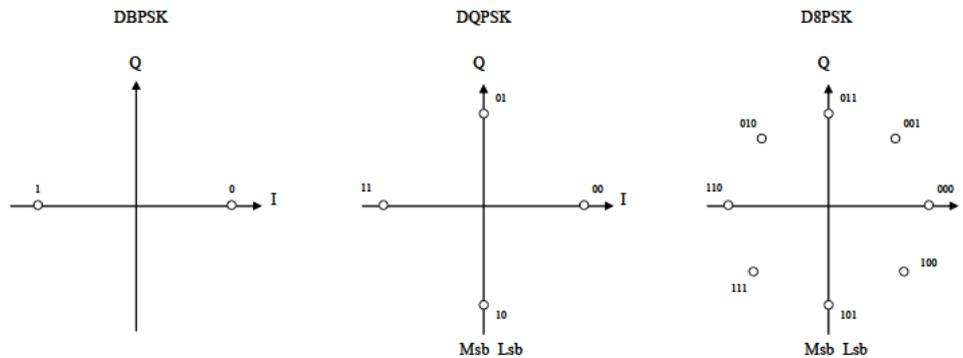


Figure 4. PRIME DPSK modulation types [12].

The most important conclusions to extract from this subsection are that PRIME uses OFDM modulation in the newly extended frequency bands ranging from 3 kHz to 500 kHz, which can be subject to noises present on the electrical power grid for those frequencies, and that different modulation types allow for a trade-off between speed and robustness depending on medium conditions.

2.2.2 MAC LAYER

The understanding of the basic concepts of the MAC layer of the PRIME protocol is critical in order to comprehend the project.

Firstly, a PRIME subnetwork is a hierarchical tree-like structure with two different types of nodes:

- **Base Node (BN):** The base node is the master node that manages a PRIME subnetwork. There is a single BN per subnetwork, and it is responsible of managing the resources and connections, which makes it the most important element.
- **Service Node (SN):** The service nodes are the remaining elements of the subnetwork. The smart meters must register into the subnetwork to become service nodes which depend of their corresponding BN.

Secondly, the service nodes can have three different states and they change automatically while the PRIME subnetwork is operating (Figure 5):

- **Disconnected:** This is the initial state of any service node before it registers into the PRIME subnetwork. While in this state, the smart meter cannot communicate its own data or switch data from other SNs and its only goal is to search for a subnetwork that it can reach and register on it.
- **Terminal:** This state allows the SN to establish new connections and transmit data, but not to switch data from other SNs.
- **Switch:** This state is the most complex one and allows the SN to establish new connections, transmit its own data and switch data from other SNs. The switching function allows some SNs in switch state to act as repeaters so that other SNs that cannot communicate with the BN otherwise can also be part of the subnetwork. Consequently, the tree topology of a PRIME subnetwork has the switches as branches that join the different levels of the subnetwork (Figure 5).

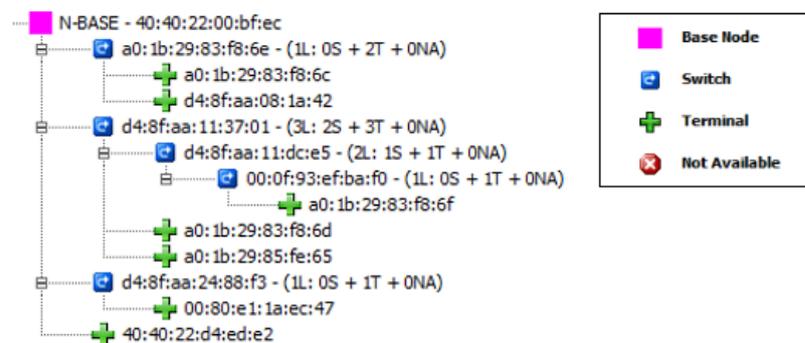


Figure 5. PRIME subnetwork topology (4 levels).

Thirdly, the transitions between one functional state to other are the following (Figure 6):

- **Registration:** It is the transition between the disconnected state and terminal. The BN includes the SN as a part of list of the registered nodes.
- **Unregistration:** It is the opposite process to the registration, and it can happen both from switch and terminal states. The BN removes the SN from the registered nodes list. Unregistration may be initiated by the SN (if it wants to find a better attachment to the subnetwork) or by the BN if the SN does not fulfill the conditions to be connected and defined by some MAC procedures.

- **Promotion:** It represents the transition from terminal to switch. The process starts when a SN that cannot directly communicate directly with the BN starts to look for an available switch. Then, a terminal that receives those messages talks with the BN and asks for the promotion to switch. Finally, the SNs that required the switching can register as terminals in the subnetwork thanks to the repeating function of the switch.
- **Demotion:** It represents the transition from switch to terminal.

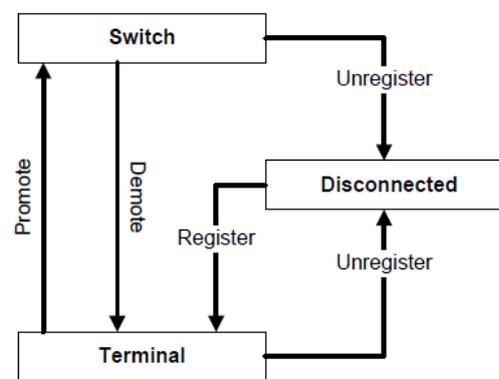


Figure 6. Service Node states and transitions [6].

In summary, the topology of a PRIME subnetwork is a tree-like structure with the BN as the root, the switches as the branches and the terminals acting as the leaves. As PRIME is a plug-and-play technology, this network is built automatically and it also dynamically autoconfigured, as the topology evolves with time. Stable networks have less disconnections and thus also remain more time at the same state, which favours PLC performance.

2.2.3 PRIME v1.4

The latest PRIME version includes some new additions that are present both at the physical (PHY) and media access control (MAC) layers of the protocol. These new upgrades seek an increase in communication robustness and an extension of the usable bandwidth with respect to the previous version of the standard, preparing PRIME for LV grid enhanced operation.

The main changes present in the PHY layer are the following [13]:

- A new type of PHY frame is defined, named Type B, while the PHY frame that was used in PRIME v1.3.6 is now named Type A. The preamble of the new type of frame lasts 8.192 ms (four times more than Type A), the header has a length of four OFDM symbols (two times more than Type A) and the maximum number of OFDM symbols that the payload can carry have been increased from 63 to 252 (Figure 7 and Figure 8). Essentially, this new frame type has been designed so that the new robust transmission modes can be supported.

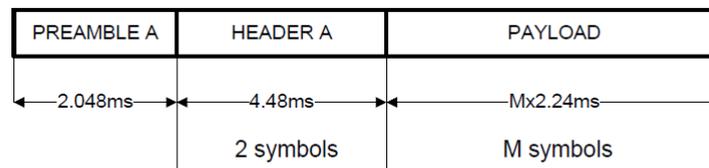


Figure 7. Type A PHY frame [13]

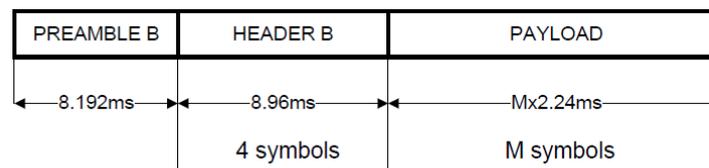


Figure 8. Type B PHY Frame [13]

- Two additional robust transmission modes have been included named: robust DPSK and robust DQPSK. The purpose behind these two new modulation schemes is to improve the resilience against impulsive and interfering noises by means of the inclusion of repetition encoding after the corresponding convolutional encoding. Besides, the existence of two robust alternatives allows for a trade-off between robustness and throughput depending on channel conditions. As it was mentioned before, PHY frame Type B is used for these modes. The comparison between PRIME v1.3.6 and v1.4 can be observed in Figure 9:

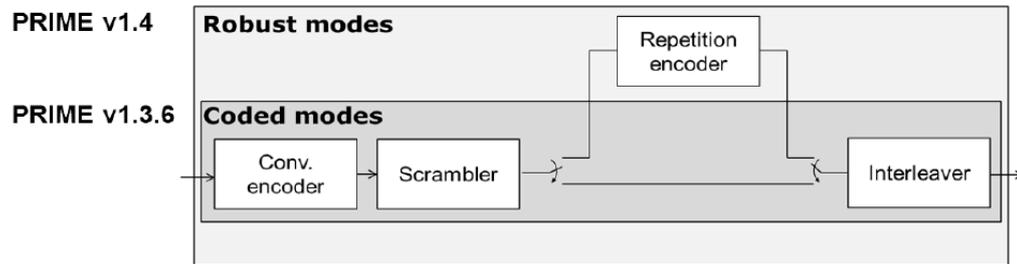


Figure 9. Block scheme of the PRIME v1.4 robust modes FEC mechanism [13]

- Frequency band extension with the addition of six new channels for data transmission, reaching a total of eight channels. This band extension multiplies by eight the bandwidth originally available, covering the FCC and ARIB bands. The new channels are in the interval between 41.99 kHz and 471.68 kHz, with guard intervals as separation of 7.3 kHz (Figure 10). This is no doubt the most important improvement in PRIME v1.4 as it does not only permit much higher data rates, but it also adds a great deal of flexibility. This flexibility comes due to the possibility of using different channels depending on channel conditions (noise avoidance) and even combining channels together to achieve even higher speeds. The combination of channels into a band is managed by the MAC layer and can achieve baud rates ranging from 5.4 kbps (one channel and the most robust mode) to 1028.8 kbps (eight channels and the least robust mode).

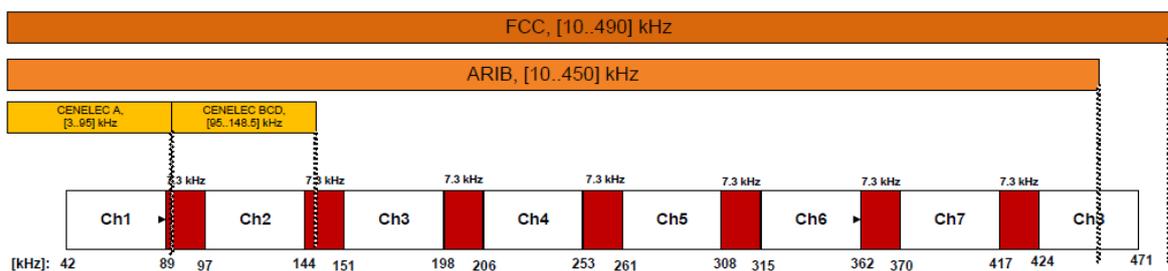


Figure 10. PRIME v1.4 frequency bands [13].

The main changes present in the MAC layer are the following:

- With respect to the MAC frame format and channel access, MAC frames can now have a length of 276, 552, 828 or 1104 symbols (decided by the BN depending on channel conditions), beacon slots and contention free area (CFP) are merged into a single CFP (the BN makes the reservations for the beacons) and there have been some minor adjustments in the CSMA-CA algorithm.
- With respect to the selection of the modulation mode, the Robustness-management (RM) algorithm has been updated to include the two new robust modes. There are a total of eight available modulation modes (Robust DBPSK, Robust DQPSK, DBPSK_CC, DBPSK, DQPSK_CC, DQPSK, D8PSK_CC and D8PSK) and the algorithm dynamically selects the most adequate one according to the link quality information carried by all generic packets.
- Multicast switching has been improved with a new mechanism.

Last but not least, the so-called Backward Compatibility (BC) mechanism has been created to enable the coexistence of v1.3.6 and v1.4 smart meters on the same network. To make it possible, a Backward Compatible PHY frame (BC PHY frame) and MAC backward compatibility features have been included (Figure 11). This mechanism is incredibly useful as it brings the possibility of having mixed networks, not requiring to fully deploy PRIME v1.4 meters in a specific network if it is not necessary or allowing to gradually deploy in stages.

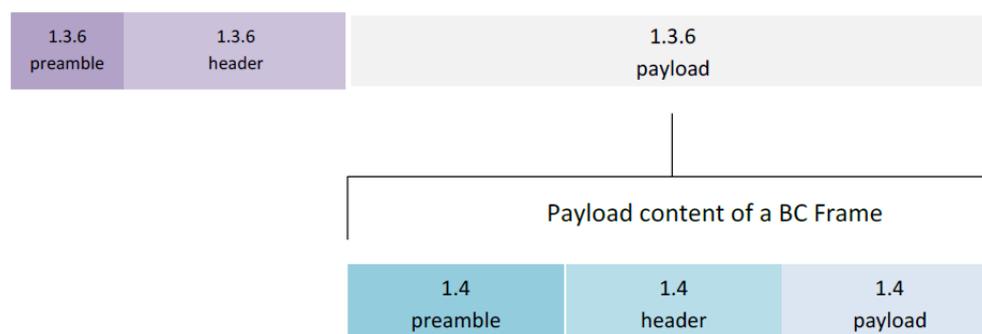


Figure 11. BC PHY frame [13].

2.3 SS CATEGORIZATION METHODS

At a certain stage of the project, a classification of the SSs into categories with similar characteristics based on the PLC performance results was done to structure the deployment. The two methods used for the categorization, Principal Component Analysis (PCA) and k-means, will be explained briefly so that the next chapters can be better comprehended.

2.3.1 PRINCIPAL COMPONENT ANALYSIS (PCA)

Principal Component Analysis is a data dimensionality reduction method normally used for large data sets when a lot of input variables are present. This method frequently used for creating a new set of reduced variables as input for machine learning algorithms.

The method looks for a new set of axes, called principal components, which are linear combinations of the original axes and in which the variance of the data is maximized. The algorithm tries to put the maximum possible information in the first component, then the maximum remaining in the second components and so on. The results if a set of n variables are fed to the algorithm are n principal components, but the most important part comes afterwards, as the variance explained by each principal component is also returned. With this information in hand, the first m principal components are picked until about 90-95% of the variance is explained, normally with 2 or 3 principal components this objective is achieved. Obviously, there is a tradeoff between the gained simplicity and reduced computational cost due to the reduction of dimensionality and the accuracy (some information is lost).

Be that as it may, PCA was not used to reduce the dimensionality of the data before feeding the k-means algorithm, as the dimensionality of the data was manageable (maximum of 18 variables), and no information wanted to be lost. Hence, the objective behind using PCA was to be able to visualize the formed clusters in 2D or 3D plots (more than three dimensions cannot be plotted) by reducing the dimensionality of the data to two or three principal components losing minimal information, checking the conformity of the clusters formed. Therefore, PCA was applied after forming the clusters to visualize the results.

2.3.2 K-MEANS

K-means is an unsupervised machine learning technique used to classify the data into clusters (groups) with similar characteristics. In contrast with supervised classification algorithms, the categories of the elements to be classified are unknown and the algorithm looks for k clusters which minimize the “error” of the formed groups.

The algorithm returns the centroids, which are the centers of the clusters formed, and the elements belonging to each cluster, which are the elements whose distance to that centroid is the minimum among the distances to all the centroids (Figure 12). Different types of distances can be used to optimize the clustering (Euclidean, cosine, cityblock, etc), but normally Euclidean distances are used.

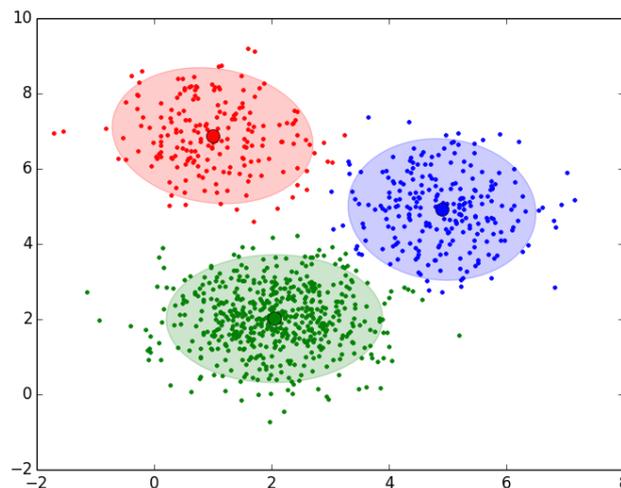


Figure 12. K-means clusters example [14].

The k-means algorithm follows an iterative process with these steps:

1. The initial k centroids are randomly placed or if it is desired the exact point of initialization can be indicated.
2. The distances of each element to the centroids are calculated and they are associated to the closest centroid.
3. The centroids are recalculated for each cluster taking into account each of its elements.

4. Steps 2 and 3 are repeated until convergence is reached.

As convergence is needed and the cluster initialization point is important, the algorithm does not guarantee convergence to the global optimum and the results depends on the initially formed clusters.

Finally, the number of clusters to be formed (k) is an input of the algorithm itself, and deciding the optimum is no easy task. When the number of clusters cannot be visualized in an obvious manner like in Figure 12, a common practice is to try the algorithm for different values of k and then compute the “error” of the formed clusters, which can be considered to be the sum of the distances of the elements to their respective centroids, and then choose the value of k . A larger k will always minimize the “error” but having more groups is suboptimal, so the recommendation is to pick the elbow of the curve (Figure 13).

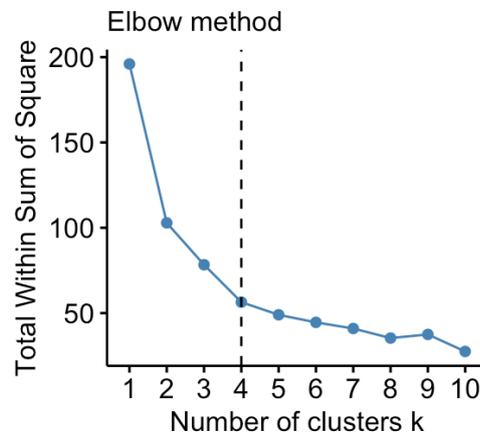


Figure 13. K-means - Optimal cluster number [15].

Chapter 3. STATE OF THE ART

The state-of-the-art research focuses on two main topics, which were both useful for understanding the subject of this study:

- **NB-PLC deployment strategies:** This is by far the most interesting topic to research, as it is exactly the field of study of this work, even in the PLC standard deployed is not PRIME v1.4. Gathering information about previous smart meter deployment methodologies that were applied and/or the results obtained would be incredibly useful.
- **NB- PLC performance:** The importance of this topic is due to the criticality of understanding which factors affect PLC performance, how and why, in order to be able to understand the results obtained and extract some meaningful conclusions.

3.1 NB-PLC DEPLOYMENT STRATEGIES

Even if smart meter deployments have not been unusual during the last decade due to the increased interest on this topic, there is a lack of literature about standard deployment guidelines that could be applicable to all or most cases. The difficulties of achieving such guidelines, are related to the challenging access to real grids by academics, the lack of electricity grid data available to the public and the fact that PLC is not such a common technology for most telecommunication experts [2].

Besides, the application of PLC technologies in the electric grid is not straightforward at all. Perhaps, the main constraint that can be found when planning a PLC deployment is the impossibility to locate network equipment at the desired optimal place as it would happen with any other telecommunication technology, as the location of the meters cannot be other than the customer premises and the concentrators must be located at the secondary substations (SSs). Furthermore, the medium used for the communication, the electric grid, is a harsh environment which exhibits constant and sometimes unpredictable changes on its

nature (line impedance, noise and frequency selective channels), severely affecting the performance of PLC systems.

For those reasons, previous knowledge of the grid is essential before any successful deployment. To make the situation even more complex, utilities have a lack of data about some segments of their electrical power grids and even inaccuracies present in their databases, as this information was not necessary to operate the grid in the traditional manner [2].

All the previously mentioned facts make ex-ante planning efforts when it comes to smart meter deployments worthless, and thus the strategy followed by Iberdrola found in [2], [9], [16] and [17] implies a post-deployment monitoring of the performance and analysis of the obtained results.

The strategy followed by Iberdrola during the first massive smart meter deployment (100,000+ meters) in Castellón included the following steps [16]:

1. Classification of the secondary substations into different categories according to the topological variables that can be predicted to have a clear influence in PLC performance. As a result, subnetworks with similar expected behavior are grouped together, creating the opportunity to find correlations between grid topology and performance results with the posterior result analysis. The chosen variables were the number of points of supply (smart meters), which affects PLC performance as the medium is shared by more meters and all of them share the same resources, and the density of points of supply (how concentrated are the meters in the customer premises), which also affects performance because it means a greater competition for the resources in an specific location and collisions may increase. Taking those two variables in mind, the SS's were categorized into 9 different categories (A to I). According to Table 1, categories A to F represent urban scenarios (very concentrated meters located in meter rooms), category G represents industrial scenarios and categories H and I represent rural or suburban environments.

Table 1. Iberdrola Castellón deployment - SS categories [16].

		Density of Points of Supply (DPoS)		
		$0 < DPoS \leq 6$	$6 < DPoS \leq 12$	$12 < DPoS$
Points of Supply (PoS)	$0 < PoS \leq 100$	Category G 151 SS's 5,380 PoS's	Category A 26 SS's 1,562 PoS's	Category B 21 SS's 1,247 PoS's
	$100 < PoS \leq 300$	Category H 76 SS's 13,752 PoS's	Category C 61 SS's 11,596 PoS's	Category D 74 SS's 13,990 PoS's
	$300 < PoS$	Category I 51 SS's 22,615 PoS's	Category E 36 SS's 15,441 PoS's	Category F 35 SS's 15,096 PoS's

2. Deployment of the smart meters and the concentrators in the largest amount as possible. The reason is that results have shown that if less than 85% of meters have been substituted, the situation is not representative of how the PRIME subnetwork will function when fully populated. As for why, if fewer smart meters are deployed there are also less opportunities for SNs to become switches and avoiding that distant meters are isolated from the subnetwork.
3. The performance assessment phase, for which Iberdrola has developed some tools that allow to measure the performance of the network in terms of its availability and stability using different metrics. These two concepts are defined as follows [16]:
 - **Availability:** If referred to the communications, it is the accumulated time (relative to the total time) for which a SN has been registered in the PRIME subnetwork. If referred to the application data, it is the ratio between successful and total meter readings.
 - **Stability:** It is defined as the ability of a SN to remain connected to the PRIME subnetwork in a consistent way. When referred to the whole subnetwork, it is the ability of the subnetwork to maintain a constant number of registered SNs.

This performance assessment phase is further separated into two stages:

- **Monitoring stage:** Stage during which data is collected both from the communications (PLC) and application (meter reading) perspective for at least one week. The first perspective provides information about how the subnetwork topology has evolved over time, accounting which SN's where connected at each time and which was their functional state (switch or terminal). The seconds perspective consists of programming the data concentrator with cycles querying all the SNs (meter reading) for periods of two days.
- **Analysis phase:** The behavior of every element of the networks is analyzed, performance results and grid characteristics are correlated, and underperforming elements can be located. The data extracted from the monitoring stage must undergo a posterior processing to generate the final variables that summarize the performance of every SS, its associated fuse boxes and every smart meter connected each of them. Table 2 contains some of the variables designed by Iberdrola to assess the performance of the deployed PRIME smart meters:

Table 2. Variables for PLC smart metering assessment [2].

Group	Subgroup	Type	Parameter
A. Subnetwork evolution	A.1 Topology stability	A.1.1 Graph	Time evolution of registered nodes: number of nodes vs. time.
		A.1.2 Value	Number of changes in the network (increment or decrement in the number of registered nodes) per minute.
		A.1.3 Set of values	Statistical values (maximum, minimum, average and standard deviation) of the availability of communications nodes measured along time.
	A.2 Application data evolution	A.2.1 Graph	Number of correct reading attempts at each cycle.
		A.2.2 Set of values	Statistical values (maximum, minimum, average and standard deviation) of successful meter readings in each cycle.
		A.2.3 Set of values	Statistical values (maximum, minimum, average and standard deviation) of the duration of successful reading attempts in each cycle.
B. Individual communication nodes	B.1 Total accumulated duration of states	B.1.1 Set of values	Total time in each of the states.
		B.1.2 Set of values	Percentage of time in each state.
	B.2 Total number of disconnections at communications level	Value	Total number of times a communications node changes from registered to non-registered state.

4. Underperforming elements are further analyzed to solve their issues and extract knowledge about what causes them, gaining expertise for future deployments.

All in all, the methodology developed by Iberdrola and the conclusions extracted from this deployment are no doubt applicable to future deployments, enabling the preparation of environments where challenges are faced in an incremental basis, allowing to fine-tune deployment arrangements (PRIME v1.4 has more adjustable parameters than the previous version) in an interoperable environment with equipment from different manufacturers. There have been no previous massive PRIME v1.4 deployments, but the knowledge gained and tools developed during previous experiences will surely be helpful for its success.

3.2 NB-PLC PERFORMANCE

All the following aspects are important when it comes to successful PLC performance and they were consequently researched. Some of the studies are just based on simulations or laboratory tests while others are based on true field cases and may be more representative. Among them, the most insightful papers are the ones that study the whole frequency spectrum that will be covered with PRIME v1.4 (3-500 kHz).

3.2.1 IMPEDANCE

The impedance shown by LV feeders depends on grid topology, type of cables used and the connected loads. Moreover, impedance also changes with the frequency of the signal that wants to be transmitted through the cables, which is especially relevant for PLC. Another concerning aspect is that not only does impedance vary with frequency but also over time, as line configuration and the loads connected by the users vary constantly, making the situation even more complex.

In urban environments, shorter cables and a big number of loads connected in parallel are common, which reduces the impedance. This makes the use of higher frequencies more appropriate, as they will not suffer as much from attenuation and are in general more resilient to noise, achieving higher SNR values. In contrast, rural environments display larger

distances and a smaller number of loads, resulting in greater values of impedance. This makes the use of lower frequencies more appropriate, as high frequencies could suffer severe attenuation [18] (Figure 14).

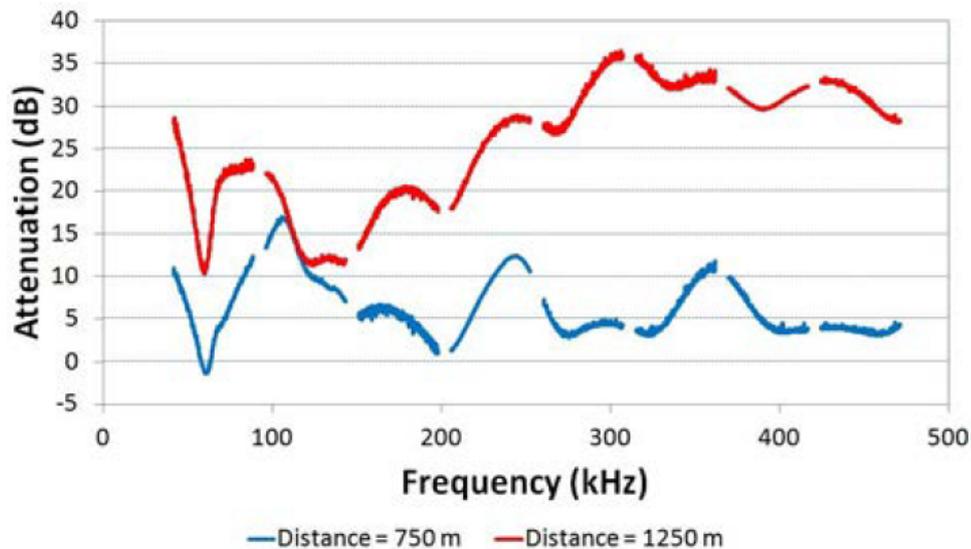


Figure 14. Attenuation in a rural feeder in the 40-500 kHz frequency range [18].

In reference [19], the importance of adequate measurement methods for LV impedance in the new frequency bands is addressed, as this aspect is essential to avoid impedance mismatches between the PLC transmitter and the electrical grid and achieving proper signal injections that can reach the receiver with high enough signal to noise ratio (SNR).

In reference [20], measurements are performed on the field for different types of feeders (urban or rural) and at different points, extracting some conclusions in the process. As agreed by most papers, the general trend of the increasing value of impedance with frequency is demonstrated (Figure 15), but this relationship is only totally true when the grid is purely inductive, which normally happens for frequencies in the 250-500 kHz range. However, the presence of resonant frequencies that cause impedance peaks disturbs this linear frequency (peak observed on about 150 kHz in Figure 15); and they are not predictable. Besides, it was also observed how the impedance of rural scenarios (Figure 16) is much more irregular than

in the urban cases, as it strongly depends on the connected devices, which are less and affect more severely as a result.

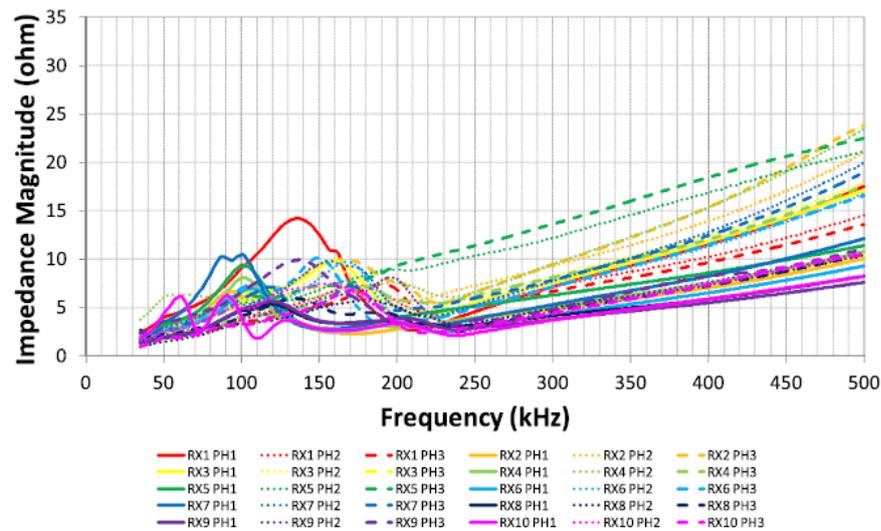


Figure 15. Impedance measurements on different points of the three phases of an urban feeder [20].

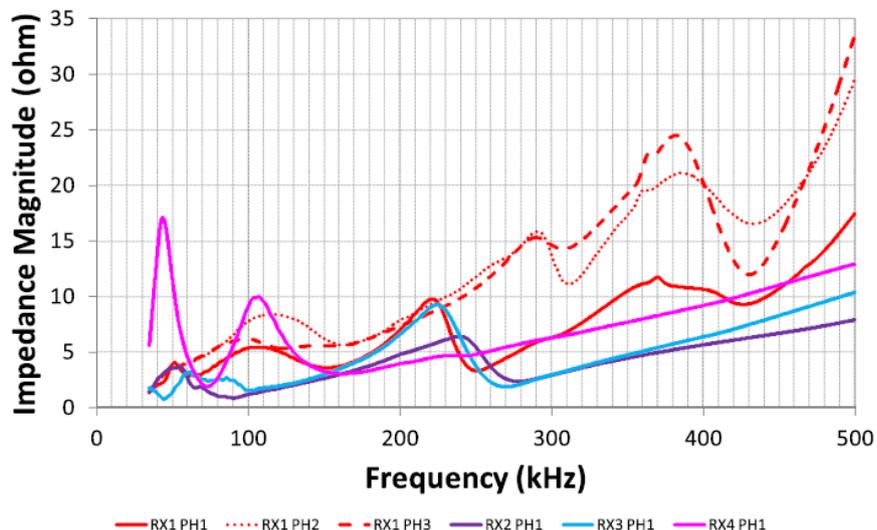


Figure 16. Impedance measurements on different points of the three phases of a rural feeder [20].

3.2.2 NOISE

Noises in PLC links can be of different types [21] and be present on different bands of the spectrum. These noises are also called non-intentional emissions (NIE) and are generated

mainly by photovoltaic inverters, battery chargers, hydropower systems, wind turbines, electric vehicle chargers, home appliances and lightning devices in the 9-500 kHz range. If they are present at the same band as the transmitted signal, they can degrade the communications or even isolate the devices depending on their level and spectral shape. The use of robust modulation and coding techniques is a possible solution or using a less affected band for communication once the NIE for the whole spectrum have been characterized.

Some papers such as references [22] and [23] are based on measurements on the field to characterize the NIE up to 500 kHz in different types of feeders. Afterwards, the NIE are classified, analysed and compared to the emissions limits for communication equipment given in EN 50065-1 [11]. This allows to assess the criticality of the different types of NIE, to hypothesise the source of each of them and to extract conclusions possibly applicable to similar feeders. Besides, the necessity to change these limits, making them stricter, to favour the performance of PLC is also mentioned

Other references such as [24] or [25] are focused on measurements on the field to characterize the emissions of different types of electric appliances or DERs. Consequently, the effects of each of them on PLC communication can be assessed. For instance, Figure 17 displays the emissions by a PV inverter, which has some sharp peaks in frequencies multiple to its switching frequency, sometimes even exceeding the allowed limits. This could result in unsuccessful PLC performance on that feeder, so the presence of solar PV generation is concerning for PLC and the use of higher frequency channels is advisable to avoid these emissions. Additionally, Figure 18 shows the emissions produced by a wind turbine were emissions that clearly exceed the limits can be observed in frequencies higher than 150 kHz, which indicates that lower frequencies would be in principle more adequate in this case.

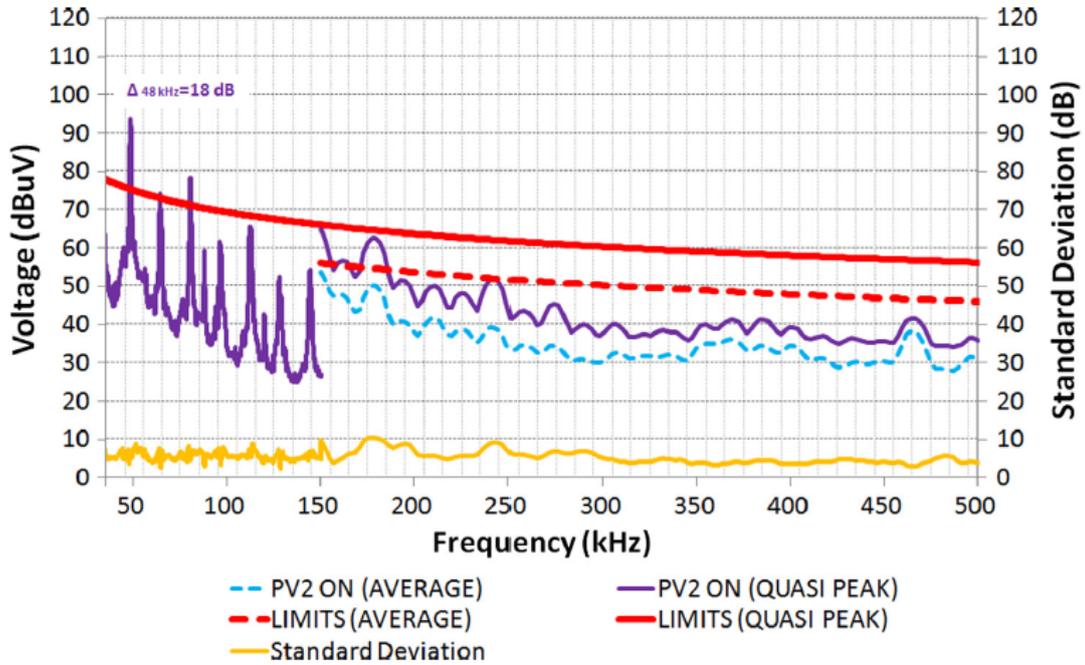


Figure 17. Non-intentional emissions generated by PV inverter vs. emission limits [24].

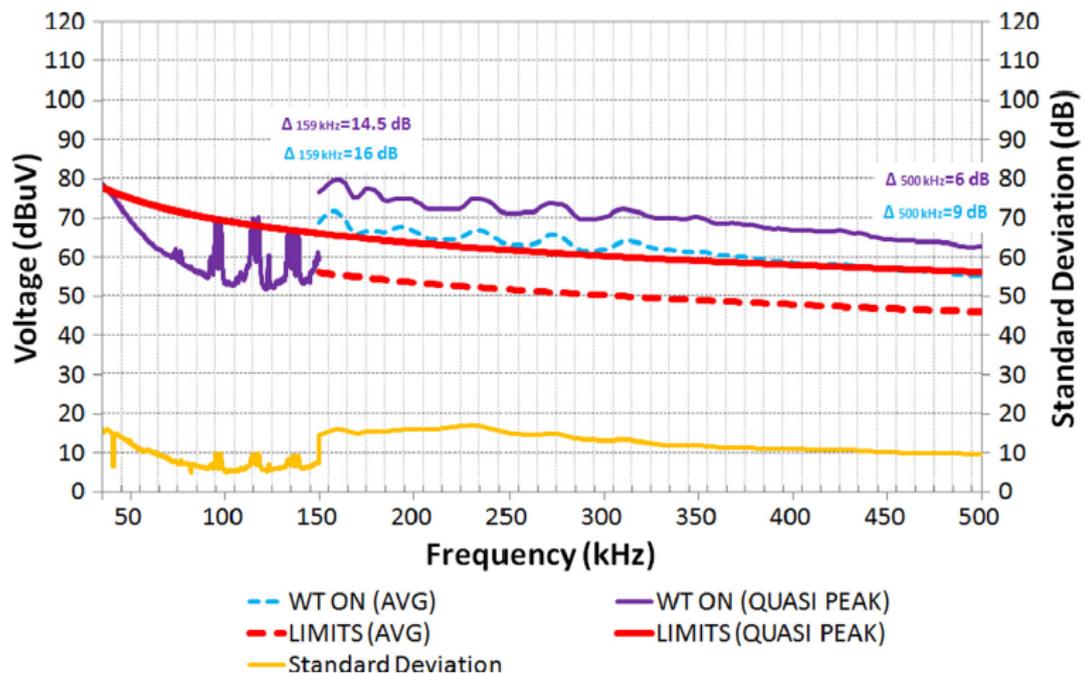


Figure 18. Non-intentional emissions generated by a wind turbine vs. emission limits [24].

3.2.3 CHANNEL SELECTION

Channel selection is key in order to be able to communicate successfully and the inclusion of new channels together with the release of the latest version of PRIME opens new possibilities.

On one hand, some studies are focused on identifying which channel is better suited for PLC communication based on some field measurements and try to look for correlations between this aspect and the topological characteristics of the grid [26] and/or noises present in the transmission medium [27].

On the other hand, other studies are purely theoretical and propose novel strategies for dynamic spectrum assignment by the base node based on link quality information [28], or even without the need of it like reference [29] proposes.

Chapter 4. PROJECT DEFINITION

4.1 MOTIVATION

The motivation behind this project is the interest of Iberdrola to perform its first massive PRIME v1.4 smart meter deployment successfully, ensuring a smooth transition from PRIME v1.3.6 to PRIME v1.4 with improved performance from the very beginning. The success of this deployment will be measured by establishing some figures of merit such as meter reading success, meter reading duration, availability or stability and making a comparison with the PRIME v1.3.6 results, expecting at least comparable performance on the SSs that had the best behavior with the previous version of the standard and a notable improvement on the rest, thanks to the new capabilities of PRIME v1.4.

Interoperability is an additional aspect to consider and of outmost importance for the utility, as environments with different mixes of smart meters from distinct manufacturers will be deployed and analyzed to assess the feasibility and performance of multi-vendor PRIME v1.4 networks on different situations. Besides, knowledge applicable to future PRIME v1.4 deployments may also be extracted if good enough results are attained, establishing some guidelines for successive massive rollouts.

In summary, the deployment of the newest version of PRIME is attractive for the utility, as it includes the major improvements mentioned in previous chapters, enhancing the capabilities of its Advanced Metering Infrastructure and enabling the development of future SG application and services.

4.2 OBJECTIVES

The main objectives of the project can be summarized as:

- Analysis of the electric topology of the SSs and first draft classification.
- Performing the large-scale monitoring of the 159 SSs, including the whole planning and adaptation of the tools to automatize the tasks as much as possible.
- Designing new variables to assess PLC performance and programming all the post-processing required to obtain them from the SS monitoring raw data.
- Categorizing the secondary substations depending on their grid topology and PRIME v.1.3.6 performance.
- Designing a strategy for the deployment, based on the PRIME v1.3.6 monitoring results and PRIME v1.4 devices arrival and testing.
- Preparing the toolkit for the PRIME v1.4 performance analysis once it is deployed on the field.

4.3 PLANNING AND ECONOMIC ESTIMATE

4.3.1 PLANNING

This subsection includes the project schedule and the description of the main tasks that were performed during the project.

The remarkable aspects from the project schedule are the following:

- The most critical part of the project is the monitoring phase, and specifically the data acquisition task. This task is a bottleneck and takes four weeks to be completed, if any shortcoming was to happen the result analysis phases would be delayed until at least most of the data is available.
- The state of the art study and the Annex B were expected to take less time, but in the end more papers about PLC performance than expected were found and some extra

time was also spent searching for more information about PLC deployment strategies.

- The project report writing task was delayed due to some issues when processing the monitoring results, which was a more critical task in comparison.
- The duration of the deployment strategy task is subject to the events related with the PRIME v1.4 equipment arrival and its testing, so deviations from the schedule are expected.

Analysis and preparation of the deployment of PRIME 1.4 smart meters in an interoperable field environment for 15,000 meters

Legend:

 Planned  Real

TASK	Plan start	Plan duration	Actual start	Actual duration	MAY				JUNE				JULY				AUGUST			
					3	10	17	24	31	7	14	21	28	5	12	19	26	5	12	19
Research					Planned				Real											
PRIME v1.4 study	05/05/2021	7	05/05/2021	5	Planned				Real											
State of the art	05/05/2021	14	05/05/2021	28	Planned				Real											
Annex B	17/05/2021	7	24/05/2021	14	Planned				Real											
Electric topology					Planned				Real											
Data acquisition	17/05/2021	4	17/05/2021	4	Planned				Real											
SS categorizing draft	20/05/2021	14	24/05/2021	14	Planned				Real											
Monitoring phase					Planned				Real											
Tool preparation	07/06/2021	7	07/06/2021	7	Planned				Real											
Data acquisition	14/06/2021	28	14/06/2021	28	Planned				Real											
Data post-processing	28/06/2021	21	28/06/2021	21	Planned				Real											
New variable design					Planned				Real											
Variable study	07/06/2021	7	14/06/2021	7	Planned				Real											
Script preparation	14/06/2021	14	14/06/2021	14	Planned				Real											
Result analysis					Planned				Real											
SS categorizing	19/07/2021	14	19/07/2021	14	Planned				Real											
Deployment strategy	26/07/2021	21	26/07/2021	20	Planned				Real											
PRIME v1.4 toolkit	09/08/2021	7	12/08/2021	6	Planned				Real											
Report writing	01/07/2021	51	12/07/2021	38	Planned				Real											

Figure 19. Project schedule.

4.3.2 ECONOMIC ESTIMATE

The whole Project has been done working remotely from home due to the COVID-19 pandemic. The only equipment required was a computer and the software tools that were used, which were all purchased by Iberdrola. Furthermore, the labor hours of a junior engineer must also be accounted, which represent the four-month internship.

Table 3 shows the economic estimate of the project:

Table 3. Project economic estimate.

Concept	Units	Nº of units	Unitary cost	Cost
Labour cost				10,200.00 €
Junior engineer	h	680	15.00 €	10,200.00 €
Depreciation				52.31 €
Laptop	h	680	0.08 €	52.31 €
Expenditures				223.00 €
Microsoft Office 365 license		1	69.00 €	69.00 €
Matlab license		1	154.00 €	154.00 €
TOTAL				10,475.31 €

The depreciation cost of the laptop has been calculated considering that it has a useful life of 5 years and that its cost is 800€:

$$Depreciation\ cost = \frac{800\text{€}}{5\ years} \cdot \frac{1\ year}{52\ weeks} \cdot \frac{1\ week}{40\ work\ h} \cdot 680h = 52.31\text{€} \quad (1)$$

Chapter 5. PROJECT DEVELOPMENT

In this chapter, a deeper description of the project is presented, and the different parts of the project are described in detail, but without analyzing the obtained results, as it will be done in the next chapter.

Apart from a more detailed project introduction, each of the sections belonging to this chapter are related to a particular objective of the project and the methodology applied on each of them is explained.

5.1 INTRODUCTION

As it was briefly explained in the introduction, the motivation for this Master Thesis arises as a result of Iberdrola's imminent first massive PRIME v1.4 smart meter rollout in Spain (about 15,000 smart meters).

In principle, the rollout will start just shortly after this Master Thesis is finished (or even later depending on when the devices are received and ready to be deployed), so the scope of this project only covers the phase previous to this massive deployment. The analysis made in this work will help to arrange the deployment in order to achieve the smoothest transition from PRIME v1.3.6, currently deployed on the field, to PRIME v1.4, something critical for the utility.

One of the obvious objectives of the rollout, being it the first PRIME v1.4 massive rollout, is to prove that PRIME v1.4 can work successfully on the field and bring notable improvements to the AMI of Iberdrola, while no degradation of performance is introduced in the network while in the transition period. Otherwise, it would make no sense to expend money and resources to substitute the PRIME v1.3.6 smart meters.

For that end, Iberdrola has carefully chosen three municipalities in Spain whose characteristics make them suitable for this first on the field “test bed”. The three municipalities are located in three different regions and for privacy reasons we will refer to them as “North”, “Central” and “East” from now onwards. Their location in the map can be observed in Figure 20:

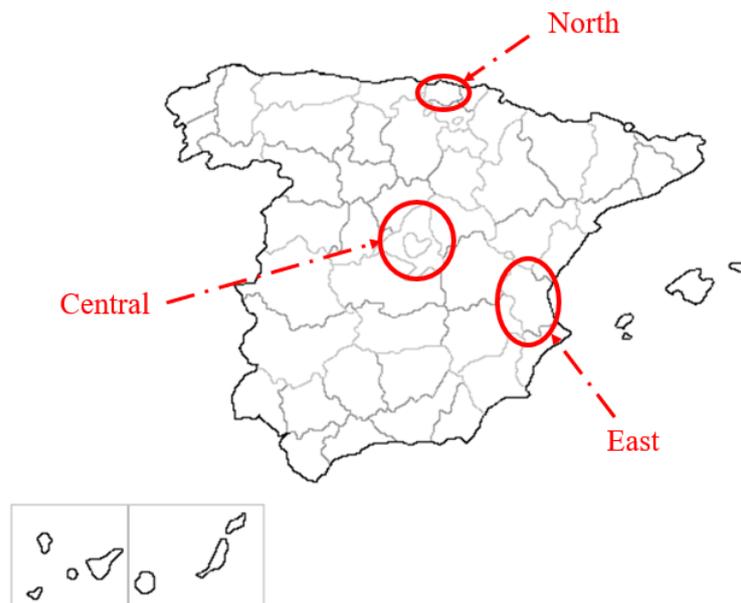


Figure 20. PRIME v1.4 deployment municipalities.

The reasons for choosing these three municipalities are the following:

- They have very similar populations (about 10,000 inhabitants), and population densities. Although East has a higher population density compared to the other two.
- The three of them have SSs of different topologic characteristics, ranging from just a few meters to hundreds of them, different kinds of SSs (underground, post, etc.) and different kinds of feeders (long or short, overhead or underground, concentrated or dispersed fuse boxes). This aspect makes them very suitable for testing PLC performance in diverse conditions.
- They are not municipalities with high populations, making the deployment easier.

Table 4 shows a summary of the SSs that have been selected for the deployment on each municipality, reaching a total quantity of 159:

Table 4. Number of SSs per municipality.

Municipality	North	Central	East
Number of SSs	55	62	42

Another additional aspect that should not be overlooked is that due to PRIME's interoperability there are concentrators and smart meters from different manufacturers deployed in the chosen SSs, whose names will not be made public due to privacy reasons. This reality has implications when it comes to the monitoring of the SSs, as it will be explained later. Table 5 summarizes it:

Table 5. Number of concentrator and smart meter manufacturers per municipality.

Municipality	North	Central	East	Global
Concentrator manufacturers	3	3	3	4
Smart meter manufacturers	6	7	7	8

With respect to the data used during the project, a clear distinction between two types of data can be easily made (Figure 21):

- **Power grid data:** This data contains the topological characteristics of the electric power grid and can be obtained from the databases of the utility using SQL queries. It is not difficult to obtain and the only complexity lies on the fact that there is a vast amount of information available and it is not easy to find the desired one. Besides, there are some errors present in the databases and it is important to check the data once downloaded.
- **PLC performance data:** This data shows how good PLC behave in a particular SS. A further distinction can be made according to the origin of the data:

- **Database data:** This data is obtained with the same procedure as power grid data and it is available at the databases of Iberdrola. However, it is very scarce and only the daily, weekly and monthly meter reading success are registered.
- **Monitoring data:** This data can only be obtained after executing the whole monitoring process of a SS, which will be explained in section 5.3 in detail. The raw data obtained in the form of hundreds of csv files per SS must be post-processed to obtain humanly understandable results. This type of data is by far the most tedious to obtain and process, but it is also the most meaningful for the project, as full knowledge of how a SS behaves can be extracted from it.

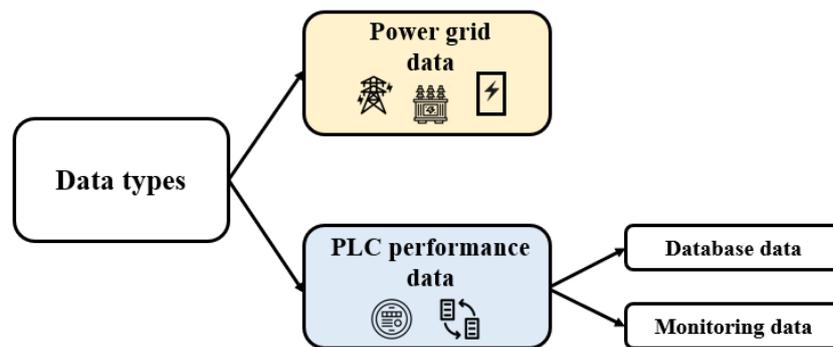


Figure 21. Project data types.

All in all, due to the large number of SSs (159), smart meters (about 15,000) and manufacturers, we can say that the project has vast amounts of information to analyze, but this Master Thesis only covers the work done to achieve the objectives that were set as part of the scope.

5.2 GRID TOPOLOGY STUDY

5.2.1 INTRODUCTION

Firstly, before talking about the grid topology study that was carried out, it is important to understand the structure of the LV segment of the electric power grid and the main elements present. Figure 22 shows the structure:

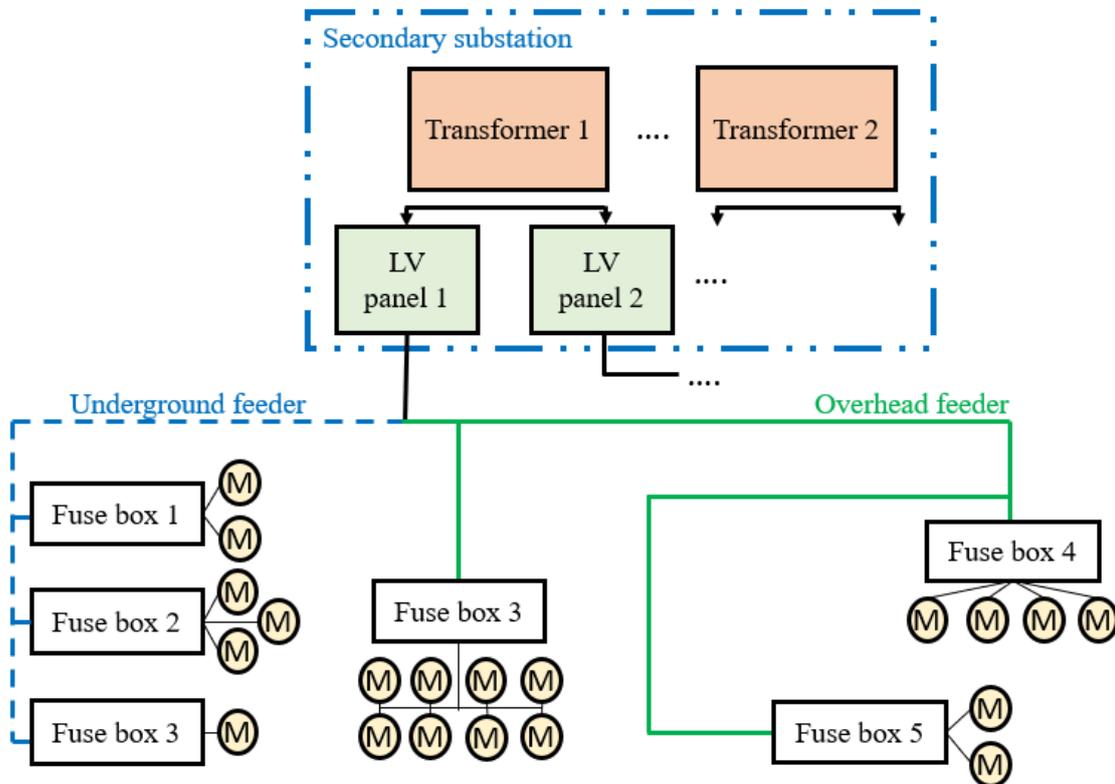


Figure 22. LV grid segment topology.

The main elements that can be found starting upstream from the SS and ending at the customer premises are the following:

- **Transformer:** This device interfaces the MV and LV sections of the grid, decreasing the voltage from MV to LV. There is normally one per SS but sometimes there are two if a lot of customers are fed with that SS.
- **Low voltage panel:** This equipment is connected to the LV side of the transformer and there can be one of them or more depending on the size of the SS. Its main functions are entry-sectioning, busbar, protection, auxiliary entry and controlling and supervising the remote management devices.
- **Feeder:** It is the line that connects the LV panels located at the SSs to the fuse boxes located close to the customer homes. They can be overhead, underground or mixed and have cables made of different types of conductor and isolation. Moreover, a LV

panel can have a single associated feeder or multiple feeders, which have branches that go to the customer homes.

- **Fuse box:** This device defines the beginning of the customer property. Its main function is the protection of the feeder that it is connected to if a fault occurs downstream. In urban environments, there is normally a fuse box (or more if the power is very high) for a whole building and in suburban environments detached houses that are nearby can share the same fuse box.
- **Smart meter:** This device is responsible of measuring the energy consumption of the customer to be able to charge the bill, as well as other functions such as voltage or current measurements or cutting the supply. In urban environments, there is normally a meter concentration room where the meters of all the neighbors are concentrated. Conversely, in rural, suburban or rural environments the meters are normally more dispersed. Furthermore, the meters can be classified according to two different aspects:
 - **Three-phase or single-phase:** Single-phase meters are the most common ones and they are only connected to one electrical phase and neutral. They allow the usage of the typical electric appliances found in a regular home. Three-phase meters are connected to the three electrical phases and allow customers to connect three phase loads such as motors and other industrial machines, they are normally related to devices with high energy consumption.
 - **Type 3, type 4 or type 5:** The only distinction between these meters is the power consumption of the customer (Figure 23). Besides, type 3 and type 4 meters are always three-phased as high energy consumptions are normally three phase loads.

Finally, it must be mentioned that Iberdrola has a GIS (Geographic Information System)-based tool to observe the grid and some of its associated elements that were mentioned beforehand, which was really handy to check the electric topology visually when odd things were visualized in the data and also to observe how the field really is as the database

variables do not give this information. Figure 24 displays a secondary substation with its associated feeders and fuse boxes, being this SS part of the PRIME v1.4 deployment.

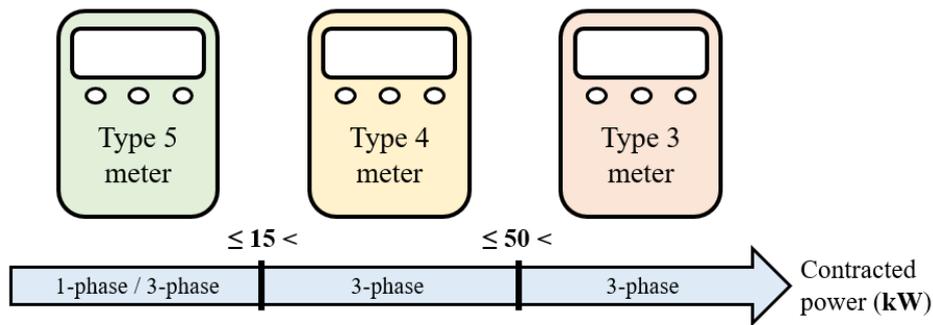


Figure 23. Smart meter types.



Figure 24. SS LV grid visualization – GENESIS.

5.2.2 ELECTRIC VARIABLE ANALYSIS

After doing some paper research and checking what information was available in the databases of the utility, the following selection of 29 variables was made (Table 6):

Table 6. Electric topology analysis variables

VARIABLE NAME	CONCEPT
SS_type	Type of SS: Post, surface prefabricated, underground prefabricated, etc.
Comms_sol	Type of communications solution of the concentrator used to communicate upstream: BB-PLC, 3G, 3GTP.
Load_type	Most common type of load: Residential homes, industrial, hotel, public lighting, etc.
N_customers	Total number of customers.
N_trif_customers	Number of customers with three-phase consumption.
%_trif_customers	Percentage of customers with three-phase consumption.
N_type3_customers	Number of type 3 customers (type 3 meters).
N_type4_customers	Number of type 4 customers (type 4 meters).
N_type5_customers	Number of type 5 customers (type 5 meters).
P_type3_customers	Contracted power by type 3 customers.
P_type4_customers	Contracted power by type 4 customers.
P_type5_customers	Contracted power by type 5 customers.
P_total	Total contracted power.
N_gen_customers	Number of customers with energy generation (DERs).
P_generation	Power generated by customers with energy generation (DERs).
Feeder_length_O	Sum of the lengths of the overhead segments of all the feeders of the SS.
Feeder_length_U	Sum of the lengths of the underground segments of all the feeders of the SS.
Feeder_length_tot	Sum of the total lengths of all the feeders of the SS.
Max_feeder_length_tot	Length of the longest feeder of the SS.
N_feeders	Number of feeders of the SS.
N_transf	Number of transformers.
N_LVpanel	Number of Low Voltage Panels.

N_fusebox	Number of fuse boxes.
N_customers_per_fusebox	Average number of clients per fuse box in the SS.
N_fusebox_per_feeder	Average number of fuse boxes per feeder in the SS.
MAX_fusebox_per_feeder	Maximum number of fuse boxes in a feeder of the SS.
Length_per_fusebox	Average feeder length between fuse boxes in all the feeders of the SS.
MAX_length_per_fusebox	Maximum average feeder length between fuse boxes in a feeder.
MAX_length_per_fusebox_GENESIS	Maximum real length between fuse boxes in a feeder (measured manually with Iberdrola's tool for LV grid visualization).

After gathering these variables, they were all arranged together in an Excel sheet and two tasks were done with this data:

- **Statistics:** The mean, standard deviation, median and quartiles were calculated for each of the variables. They were calculated firstly by considering all the 159 SSs and later by separating them by their municipalities to observe the differences. With the results, different box plots were created to visualize them.
- **Correlation analysis:** After scaling the data, the correlation matrix to see the dependence between each pair of variables was obtained using a Python script (Script 1). With this information, the redundant variables that do not add extra information could be found out. It is worth mentioning that the qualitative variables (SS_type, coms_sol and load_type) could not directly participate in the correlation matrix as the method used requires quantitative variables. Figure 25 displays an 8x8 excerpt of the 26x26 correlation matrix obtained, where for example the high correlation between the total number of customers (N_customers) and the type 5 customers (N_type5_customers) can be easily observed:

	N_customers	N_trif_customers	%_trif_customers	N_type3_customers	N_type4_customers	N_type5_customers	P_type3_customers	P_type4_customers
N_customers	1,00	0,59	-0,44	-0,08	0,19	1,00	-0,07	-0,05
N_trif_customers	0,59	1,00	0,09	0,23	0,73	0,56	0,19	0,24
%_trif_customers	-0,44	0,09	1,00	0,38	0,30	-0,45	0,32	0,18
N_type3_customers	-0,08	0,23	0,38	1,00	0,33	-0,10	0,93	0,21
N_type4_customers	0,19	0,73	0,30	0,33	1,00	0,16	0,25	0,44
N_type5_customers	1,00	0,56	-0,45	-0,10	0,16	1,00	-0,08	-0,06
P_type3_customers	-0,07	0,19	0,32	0,93	0,25	-0,08	1,00	0,16
P_type4_customers	-0,05	0,24	0,18	0,21	0,44	-0,06	0,16	1,00

Figure 25. Electric topology correlation matrix - 8x8 excerpt

5.2.3 DRAFT SS CATEGORIZATION

From the 29 electrical variables that were analyzed, the number of customers (N_customers), the average customers per fuse box (N_customers_per_fusebox) and the average feeder length between fuse boxes (Length_per_fusebox) were selected to classify the SS into groups with similar characteristics (categories) and thus possibly similar PLC behavior. The number of variables was selected to arrange SS categories that can be easily represented in a table and are humanly understandable (not obtained by means of an opaque algorithm).

Furthermore, an additional variable related to PLC performance was added to check the homogeneity of the groups after creating the categories and analyzing if the formed categories made sense. The selected variable is of the database type (explained in 5.1), is called %_success_S05 and it measures the average percentage of success when reading the smart meters of the SS once a day after a number of consecutive tries.

The tasks performed to obtain the draft SS categories can be separated into three groups:

- **Classification variable selection:** Different options were tried thanks to a Python script that made the process almost instantaneous (Script 2). The correlation analysis was used to choose these variables.
- **Classification ranges selection:** The ranges of the selected variables to set the limits between categories had to be chosen. The previously calculated statistics (mean and quartiles) were used to decide where to put the limits to obtain groups of adequate size. This part was also automatized with the help of the previously mentioned script.

- **Analyzing the results:** The size of the groups and the mean and standard deviation of the variable `%_success_S05` were checked to see if the SSs on each of them behaved similarly. This was an iterative process which meant repeating the previous tasks until obtaining a meaningful classification.

5.3 *PLC PERFORMANCE MONITORING*

In order to have full knowledge about the PLC performance of a SS, the full monitoring process must be performed. This step is critical because the variables related to average smart meter reading success that can be easily downloaded from the databases of the utility are not enough to comprehend the behavior of a SS.

The full monitoring process that was applied during the project was developed by Iberdrola previous to their first PRIME deployments some years ago and produces hundreds of csv files that are later processed in order to produce an Excel file that has very complete information about how a SS behaves.

The monitoring of a SS can be separated into three main parts, and each of them produces independent csv files:

- **Topology:** It registers how the PRIME topology evolves for one week. The service nodes dependencies, functional states (switch, terminal, etc.) and the durations and times of these states are recorded. It must be activated and deactivated. It analyses PRIME from the communications perspective.
- **Short polling cycles (Short cycles):** Constant polling cycles asking the smart meters low-load data (single measurements). The success and duration of these cycles are measured for two days. It only requires activation and deactivates automatically after two days. It analyses PRIME from the application data perspective.
- **Long polling cycles (Long cycles):** Constant polling cycles asking the smart meters high-load data (string with eight measurements). The success and duration of these cycles are measured for two days. It only requires activation and deactivates

automatically after two days. It analyses PRIME from the application data perspective.

The typical chronology of the monitoring of a SS can be observed in Figure 26:

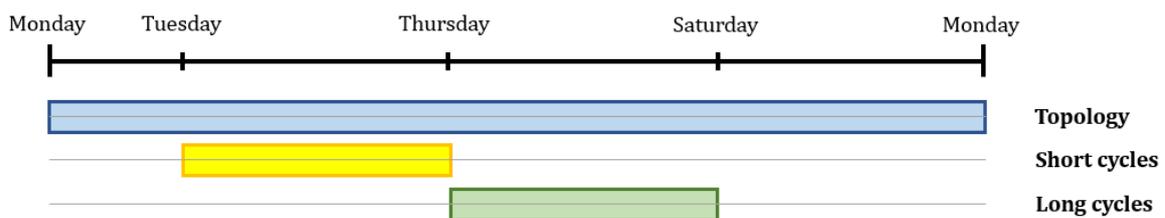


Figure 26. Typical SS monitoring chronogram.

Moreover, if significant monitoring results want to be attained, during the monitoring phase the daily tasks of the concentrator must be deactivated so that they do not affect the results. For instance, one of these tasks is to read and send the energy consumption of the customer to the utility on a daily basis. Suspending this task for a week is not such a big problem as it may seem because smart meters can store this information for about a month and in to bill the customer with a successful reading once a month is enough.

Be that as it may, in order to suspend these tasks, permission had to be asked and as it was not allowed to monitor the 159 SS at the same time, a planning separating the SSs into six groups had to be made. The planning for the monitoring of the 159 SSs can be observed in Figure 27 and the criteria to make the groups and the planning was the following:

1. It was not possible to monitor all the SSs of each municipality simultaneously, so each municipality was separated into two groups, accounting to a total of six groups.
2. Two groups can be done in parallel but not more because it would imply too much time dedicated daily to monitoring purposes.
3. The monitoring of the concentrators of one of the manufacturers (which accounts for 122 of the 159 SS) can be more automatized than the others. For that reason, the

5.4 POST-PROCESSING AND NEW VARIABLE DESIGN

5.4.1 POST-PROCESSING

As it was mentioned beforehand, the raw data that is acquired after the full monitoring process is a sequence of hundreds of csv files (topology, short cycle and long cycle files) per SS that do not give useful insight at first sight. With this information in hand, a post-processing is required to calculate high-level PLC performance variables that indicate how the technology behaves from different perspectives.

The methodology applied to process this data had also been previously developed by Iberdrola for its first PLC deployments and it calculates the variables that the utility traditionally has used to analyse PLC performance. Be that as it may, the decision to include some newly designed variables to analyse PLC performance and improve the existing approach, being some of them PRIME v1.4 specific, caused the need to adapt the already existing processing tools.

In a summarized way, apart from a slight adaptation of the existing tools, an additional posterior processing step using Python was developed from scratch to calculate these new variables (Script 3) and to concentrate all the information into a single Excel file (Script 4), allowing for an easy comparison between the SSs.

The detailed post-processing approach is presented in Annex III: Monitoring Data Post-Processing.

5.4.2 NEW VARIABLE DESIGN

The newly designed PLC performance variables were decided together with the workmates in Iberdrola with the objective of gathering as much useful information of each SS as possible, improving the already existing set of variables available to analyze PLC behavior.

Accounting the new and old ones, a total of 48 variables were calculated per SS from the monitoring data. The chosen variables were organized into different topics and concepts

depending on their nature, which gives them different objectives. Besides, these variables can be of different types, with the possibility of just being a single numerical value, a set of values or even a graph.

The complete summary of the variables, including their individual definitions, can be found in Table 7 (the new variables have been highlighted in blue):

Table 7. PLC performance analysis variables - New variables (blue).

Topic	Concept	Id	Type	Definition
A. Subnetwork evolution	A.1. Topology stability	A.1.1	Graph	Time evolution of registered nodes and switches.
		A.1.2	Value	Presence of periodic noise.
		A.1.3	Set of values	Statistical values (maximum, minimum, average and standard deviation) of the registered nodes.
		A.1.4	Set of values	Statistical values (maximum, minimum, average and standard deviation) of the switches.
	A.2. Meter reading - Short cycles	A.2.1	Graph	Time evolution of percentage of successful meter readings at each cycle.
		A.2.2	Set of values	Statistical values (maximum, minimum, average and standard deviation) of successful meter reading percentage.
		A.2.3	Graph	Time evolution of duration of meter readings at each cycle.
		A.2.4	Set of values	Statistical values (maximum, minimum, average and standard deviation) of the duration of meter readings.
	A.3. Meter reading - Long cycles	A.3.1	Graph	Time evolution of percentage of successful meter readings at each cycle.
		A.3.2	Set of values	Statistical values (maximum, minimum, average and standard deviation) of successful meter reading percentage.
		A.3.3	Graph	Time evolution of duration of meter readings at each cycle.
		A.3.4	Set of values	Statistical values (maximum, minimum, average and standard deviation) of the duration of meter readings.
B. Subnetwork availability	B.1. Communications availability	B.1.1	Set of values	Communications availability (percentage of time) per meter concentration or meter.
		B.1.2	Set of values	Statistical values (maximum, minimum, average and standard deviation) of the communications availability of the whole subnetwork.
	B.2. Meter availability - Short cycles	B.2.1	Set of values	Meter availability (percentage of time) per meter concentration or meter during short cycles.
		B.2.2	Set of values	Statistical values (maximum, minimum, average and standard deviation) of the meter availability during short cycles of the whole subnetwork.
	B.3. Meter availability - Long cycles	B.3.1	Set of values	Meter availability (percentage of time) per meter concentration or meter during long cycles.
		B.3.2	Set of values	Statistical values (maximum, minimum, average and standard deviation) of the meter availability during long cycles of the whole subnetwork.
B.4. Compared availability		Graph	Representation comparing B.1.1, B.2.1 and B.3.1.	
C. Subnetwork topology	C.1. Most common topology state	C.1.1	Graph	Topology representation.
		C.1.2	Value	Number of switches.
		C.1.3	Value	Number of switches in level 1.
		C.1.4	Value	Number of switches in level 2.
		C.1.5	Value	Number of terminals.
		C.1.6	Value	Number of switching levels.

		C.1.7	Value	Level with the most meters. Histogram of the number of meters with a percentage of time in their most common topology state. Histogram with the percentage of meter with a percentage of time in their most common topology state.
		C.1.8	Graph	
		C.1.9	Graph	
	C.2. <i>Instantaneous topology state</i>		Graph	Instantaneous topology representation, to observe specific instants.
D. <i>Individual nodes</i>	D.1. <i>State duration</i>		Set of values	Percentage of total time in each state (switch, terminal, disconnected).
	D.2. <i>Number of disconnections</i>		Value	Number of times a communication node changes from registered to non-registered state.
E. <i>Inter-operability</i>	E.1. <i>Concentrator information</i>	E.1.1	Name	Name of the manufacturer of the concentrator.
		E.1.2	Value	Firmware version of the concentrator.
	E.2. <i>Meter information</i>	E.2.1	List	Names of the manufacturers of the meters.
		E.2.2	Set of values	Number of connected meters from each manufacturer.
		E.2.3	Set of values	Percentage of meters from each manufacturer
	E.3. <i>Meter availability</i>	E.3.1	Set of values	Average communications availability for each manufacturer.
		E.3.2	Set of values	Average meter availability for each manufacturer during short cycles.
		E.3.3	Set of values	Average meter availability for each manufacturer during long cycles.
	E.4. <i>Topology</i>	E.4.1	Set of values	Number of switches from each manufacturer during the most common topology state.
		E.4.2	Set of values	Number of terminals from each manufacturer during the most common topology state.
		E.4.3	List	Switch-terminal relationships manufacturer combinations for each topology level.
	F. PRIME <i>tunable parameters (only for PRIME v1.4)</i>	F.1. <i>SAR size</i>		Value
F.2. <i>Frame length</i>			Value	Selected frame length.
F.3. <i>Modulation robustness management</i>			Value	Selected modulation robustness management mode.
F.4. <i>Beacon frequency</i>			Value	Selected frequency for beacon transmission and reception.
F.5. <i>Alive mechanism type</i>			Value	Selected alive mechanism.
F.6. <i>Channel/s</i>			Value	Selected frequency band for communications.

The six different topics that the variables can belong to are the following [15]:

- A. Subnetwork evolution:** They provide information about the vision the BN has when controlling the subnetwork both from the communications (A.1) and application (A.2 and A.3) perspective.
- B. Subnetwork availability:** They provide information about the plausibility of accessing the meters of the subnetwork (availability) both from the communications (B.1) or application (B.2 and B.3) perspective. Besides, B.4 provides a comparison between the communications and meter reading availabilities.

- C. Subnetwork topology:** They register the PRIME hierarchy during the whole monitoring and its evolution, also registering the most common state. Moreover, eight new variables related to the network hierarchy existing during the most common state have been added.
- D. Individual nodes:** They measure the behavior from the communications perspective of each individual SN. The duration and changes between the functional states are registered.
- E. Interoperability:** They segment the information depending on the smart meter manufacturer. Some of the variables from previous topics are used again but segmented (availability), while new ones have also been created such as the number of meters from each manufacturer or how is the subnetwork hierarchy organized. They can be very useful to analyze the performance of interoperable environments.
- F. PRIME tunable parameters:** They are all new and are related to some new tunable options that PRIME v1.4 introduces and that make the new version of the standard more versatile. Registering how these parameters were set on the deployed subnetworks will help to later identify which ones work better on each environment, something currently unknown.

5.5 SS CATEGORIZING

After concentrating all the high-level information of the monitoring results into a single Excel file, the objective was to use some of those variables to form groups of SSs with similar characteristics. This practice can be very useful for organizing the PRIME v1.4 deployment, structuring which SSs must be deployed on each incremental stage.

The methodology applied to form the SSs groups was a clustering technique named k-means, previously explained in 2.3.2.

5.5.1 CLUSTERING VARIABLES

Five different variables were chosen for the clustering after discussing it together with the team. These variables are related to the availability and stability of the PRIME subnetwork of each SS and its size. The five variables chosen are of the type “set of values” and thus a maximum of 18 variables per SS are available for the clustering.

5.5.1.1 Connected nodes (A.1.3)

The average and standard deviation values have been chosen because they are the most significant. The average value of the connected nodes gives information about the size of SS (number of smart meters) and the standard deviation gives some information about its stability, as high values of standard deviation characterize unstable subnetworks with a probability of a noise being present.

For an easier comprehension, variable A.1.1 is the visual representation of A.1.3 and it is presented in Figure 28, Figure 29 and Figure 30. Figure 28 shows a very stable subnetwork, while Figure 29 shows a subnetwork with a periodic noise and Figure 30 shows a network with a random patterned noise. Noises result in a high value of the standard deviation. The tunable parameters and the different channels available with PRIME v1.4 will no doubt help to improve the avoidance of these noises using new channels.

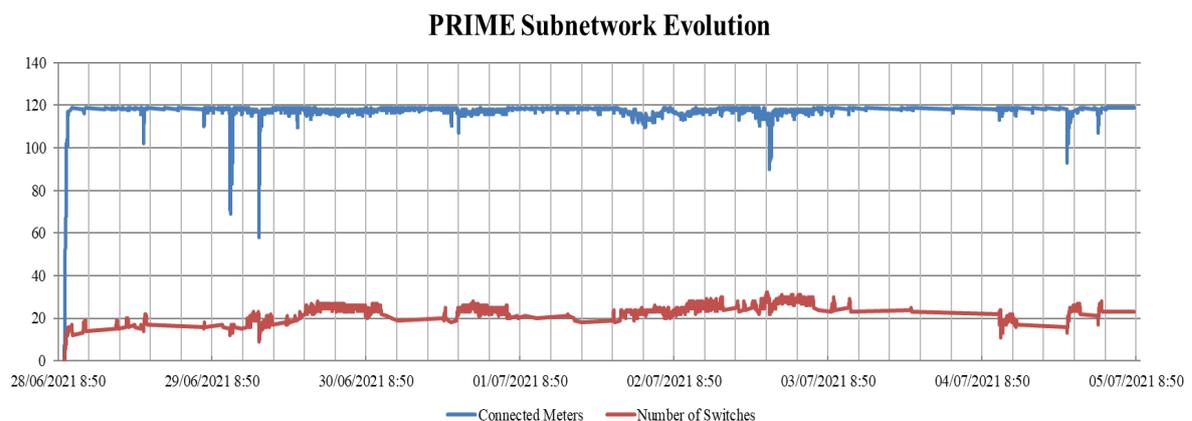


Figure 28. Connected nodes and switches (A.1.1) - Stable network.

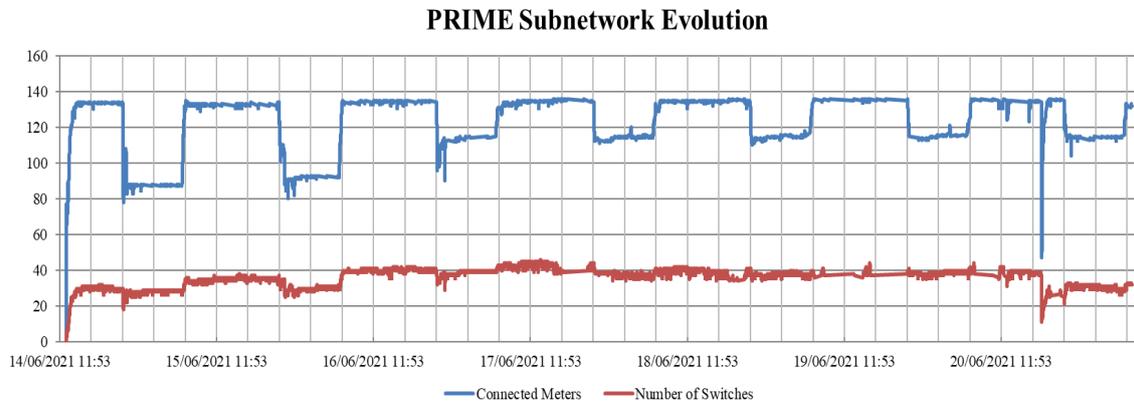


Figure 29. Connected nodes and switches (A.1.1) - Periodic noise.

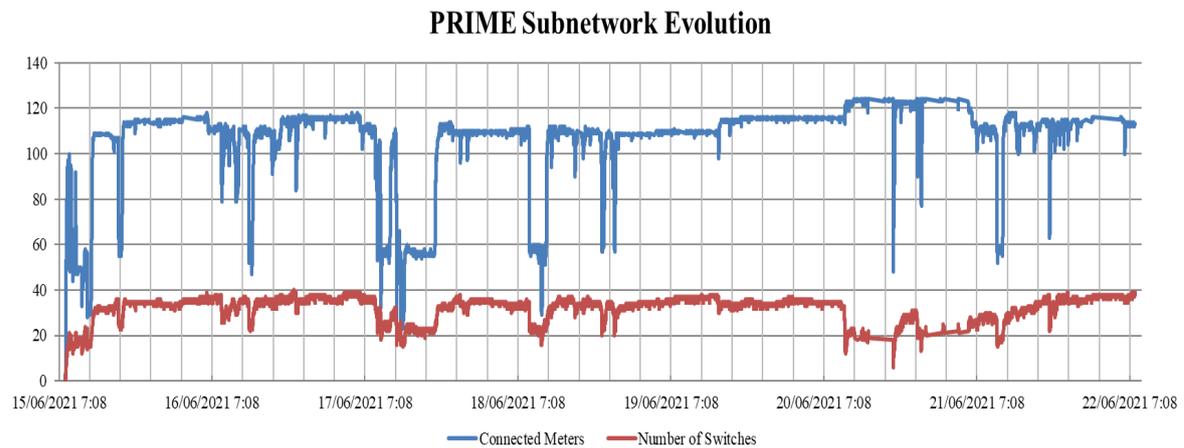


Figure 30. Connected nodes and switches (A.1.1) - Random patterned noise.

5.5.1.2 Communications availability (B.1.2)

Also called “PRIME availability”, it measures the availability of the PRIME subnetworks from the communications perspective, it being the percentage of time that the SNs are in the registered functional state. The chosen values have also been the average and the standard deviation as they are the most significant. The average provides overall information and the standard deviation shows if there are some smart meters with atypical performance present in the subnetwork, indicating the possibility of the presence of a noise in some parts of the grid.

5.5.1.3 Meters availability – Short cycles (B.2.2)

It measures the availability of the subnetwork from the application perspective, that is, the meter reading success during the short polling cycles. The chosen values have been the average and the standard deviation once again due to the previously mentioned reasons.

5.5.1.4 Meter availability – Long cycles (B.3.2)

It measures the availability of the subnetwork from the application perspective, that is, the meter reading success during the long polling cycles. The chosen values have been the average and the standard deviation once again due to the previously mentioned reasons.

5.5.1.5 Stability histogram (C.1.9)

This is the only variable from the selected ones that is new and has not been traditionally used by Iberdrola, so this is an innovation that has been applied. The variable gives highly detailed information about the stability of the PRIME subnetwork, so it was decided to include it in the SS categorizing process as it seemed to be a promising addition.

Basically, during the operation of a PRIME subnetwork, each SN is part of the PRIME hierarchy and can depend of different switches (be on a different branch of the topology) and have different functional states (terminal, switch or disconnected). Each of these states that happen during the monitoring, taking into account both mentioned aspects, is registered together with its time duration, being the “most common topology state” the one that accumulates more time during the monitoring. After obtaining this “most common topology state”, the percentage of time respect to the total monitoring time that each smart meter has spent on this state can be calculated. Finally, 10 stability ranges have been created (100-90%, 90-80%, 80-70%, ...,10-0%) and with the percentage of meters of the SS that fall on each range a stability histogram variable (C.1.9) has been created. The representation as a histogram is very useful because it is a curve with a shape, which can be more easily interpreted visually than 10 numerical values.

For a better understanding of this variable, the histograms corresponding to four SSs with different stability patterns will be shown (Figure 31):

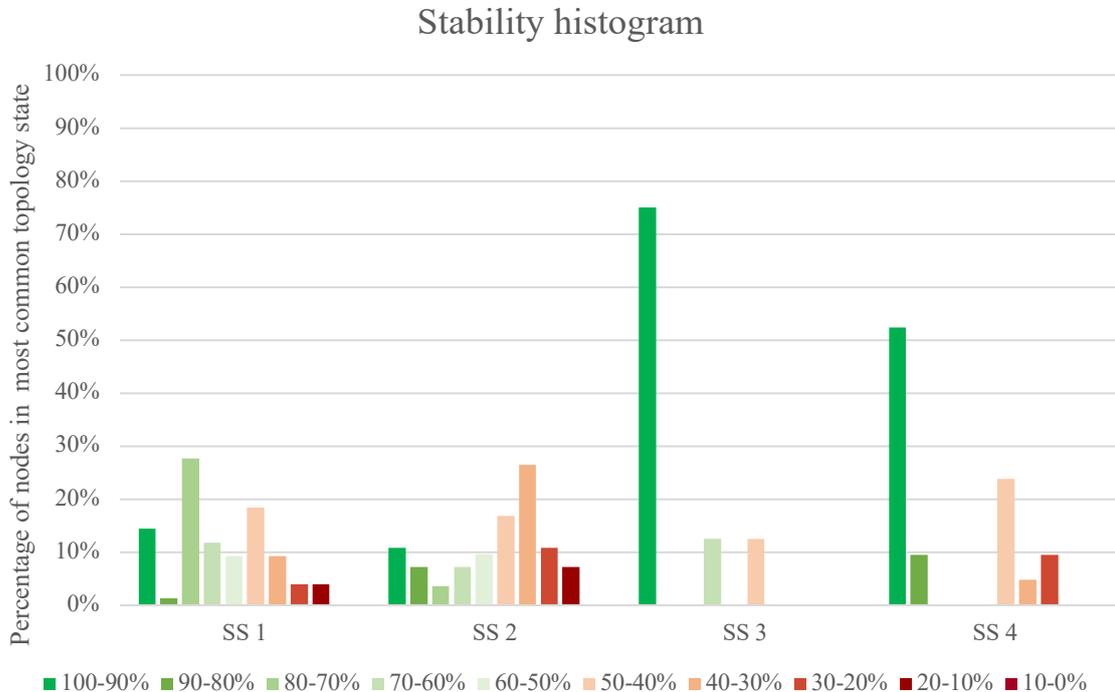


Figure 31. Stability histogram (C.1.9) - 4 SS comparison.

As it can be observed, SS 3 is the most stable of them, having more than 70% of its meters with their stability in the best range (90-100%). When it comes to SS 4, it is a bit less stable than SS 3 as it has most of its meters in the best range but has others that are unstable, even if we also consider it as a stable SS. With respect to SS 1, it has most of its meters in the middle, so the peak of the curve is centred, which results in a somewhat stable SS. Finally, SS 2 has the peak of the curve moved to the right side, which indicates that most of its meters are unstable and that will be the overall behaviour of the SS.

The developed SS summary Excel sheet (explained in Annex III: Monitoring Data Post-Processing) has the option to select the desired SSs from the list and plot the histograms together, providing the possibility of comparing the curves (Figure 31). Additionally, if the “change rows for columns” option is selected, the curves can be overlapped, allowing the identification of SSs with similar stabilities (Figure 32), something very useful for categorization purposes.

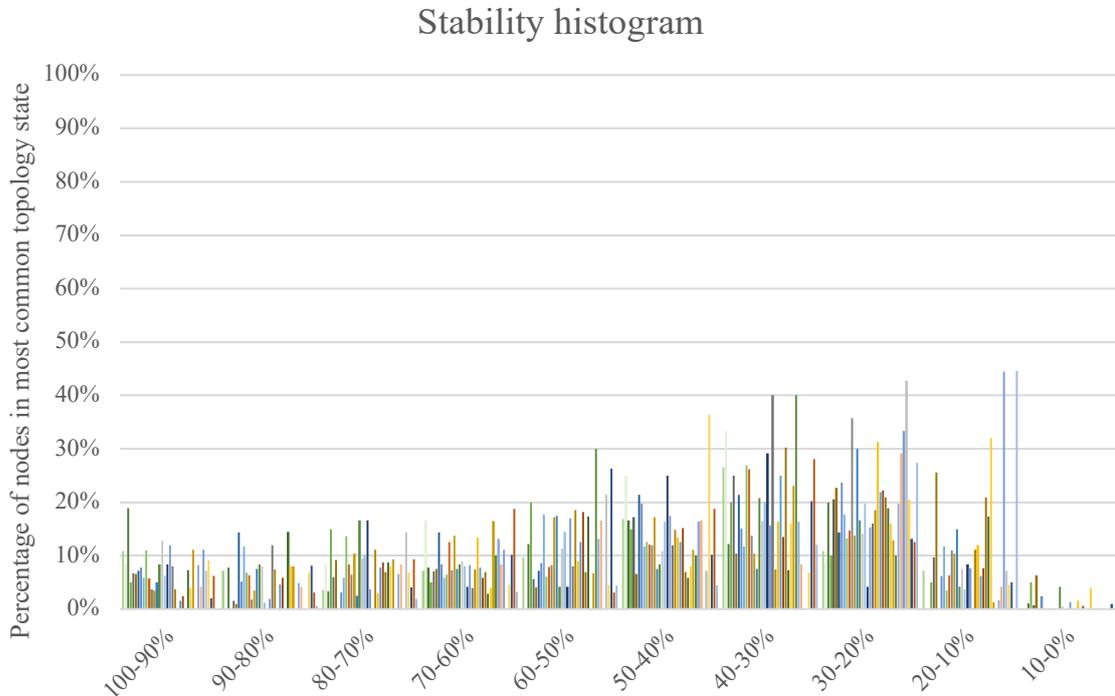


Figure 32. Stability histogram (C.1.9) – Unstable SSs

For instance, Figure 32 displays a group of SSs with bad stability, as the shape of their curves has the peak located at the right side, having most meters located in some of the least stable ranges. Conversely, Figure 33 shows a group of SSs with very good stability, as the shape of their curves has the peak located at the left side, having approximately the 50% of the smart meters in the most stable range and Figure 34 shows a group of SSs with average stability, with the peak located in the middle of the curve.

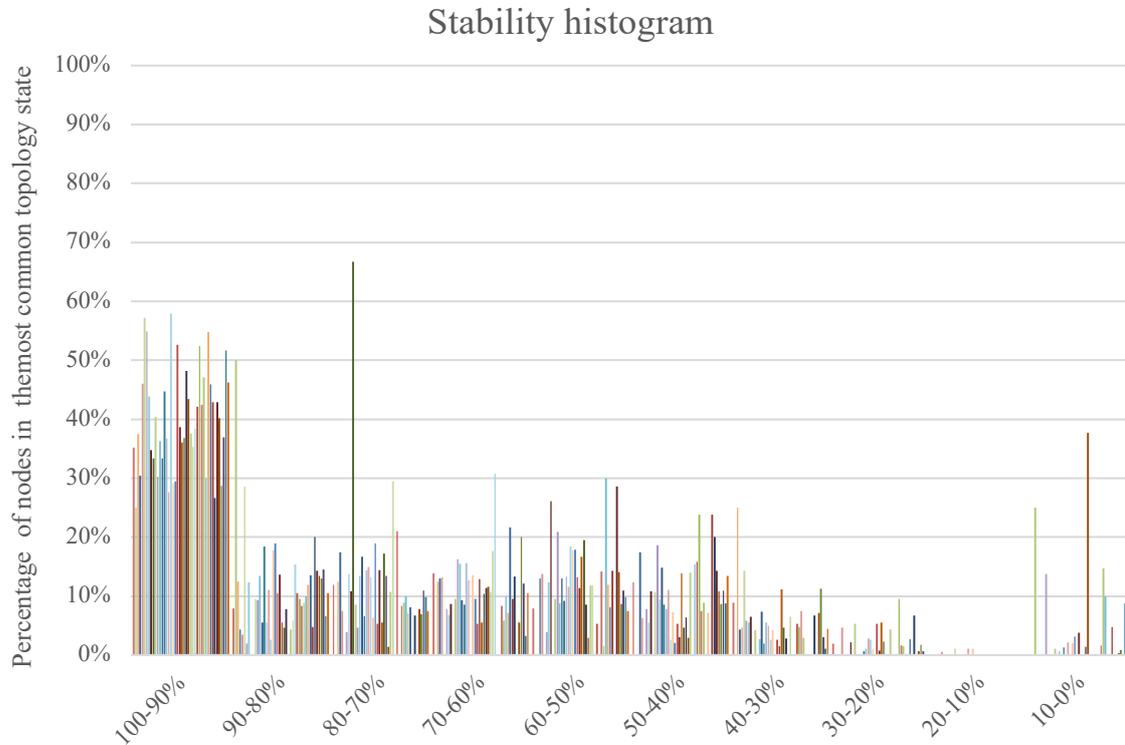


Figure 33. Stability histogram (C.1.9) – Very stable SSs.

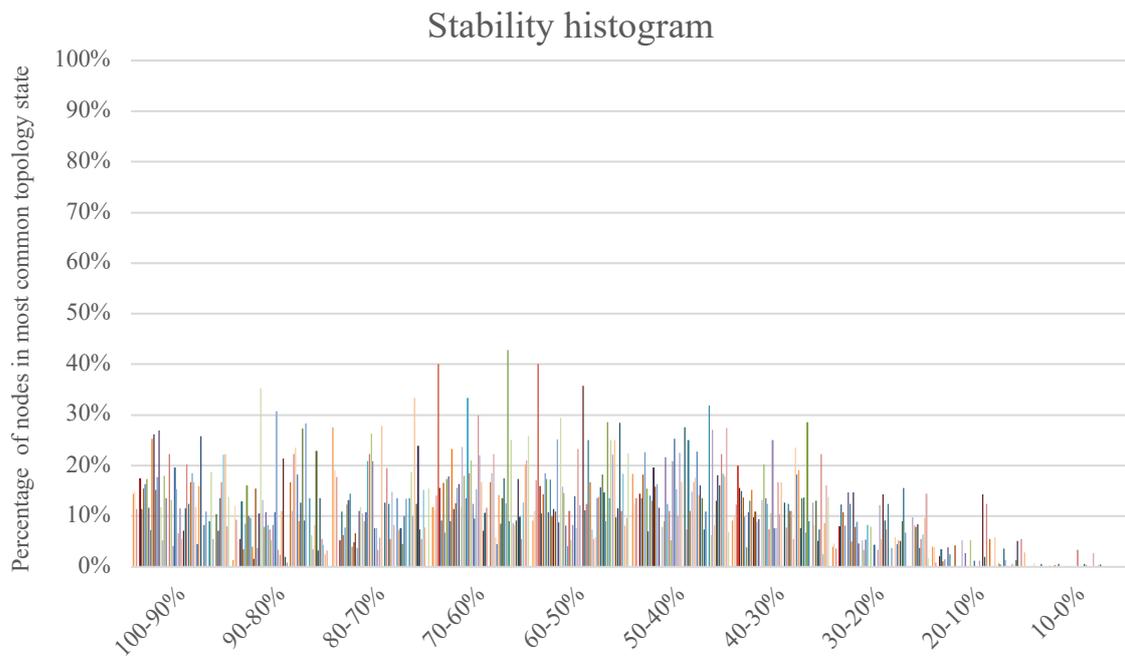


Figure 34. Stability histogram (C1.9) – SSs with average stability.

5.5.2 K-MEANS CLUSTERING

The method applied for grouping the SSs has been the unsupervised Machine Learning method named k-means. The method allows to form clusters (groups) of elements considering their similarities in the input variables. The details of k-means were previously explained in 2.3.2.

Besides, as five different variables were chosen, the idea was to try different combinations and select the most optimal according to the obtained results. A total of five different input variable combinations were tried:

- **Option 1:** Stability histogram (C.1.9).
- **Option 2:** Availabilities (B.1.2, B.2.2 and B.3.2).
- **Option 3:** Connected nodes (A.1.3) + Availabilities (B.1.2, B.2.2 and B.3.2).
- **Option 4:** Stability histogram (C.1.9) + Availabilities (B.1.2, B.2.2 and B.3.2).
- **Option 5:** All of them (A.1.3, B.1.2, B.2.2, B.3.2, C.1.9).

The methodology applied for each of the five options, which used a MATLAB script (Script 5) and Excel dynamic tables from the summary excel file consisted on the following steps:

1. Selecting the option (1 to 5) that wants to be tried by changing the input parameter from Script 5 that selects the adequate variables to be downloaded from the summary Excel file.
2. Selecting the number of clusters that wants to be tried. To ease this process, the MATLAB script plots the sum of the centroid distances vs. the number of clusters (k). The idea is to see how much improvement is attained with each extra group and selecting the optimal value of k (Figure 35). In this example the optimal number of groups would be between 5 and 8.

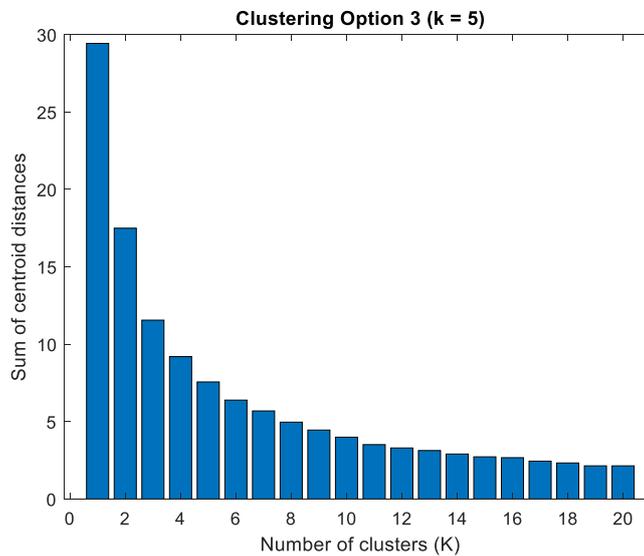


Figure 35. Cluster number selection – Option 3.

3. Running the k-means algorithm and seeing how many SSs fall on each group. If many small groups appear, the idea of going back to step 2 and reducing the number of clusters comes to the mind. If it is still not clear enough, the decision made in some cases was to separate the option into two alternatives with a different number of clusters and completing the whole analysis for both cases.

4. Applying Principal Component Analysis (PCA) to the results so that the different clusters can be visualized on 2D or 3D plots (more than 3 variables cannot be plotted and understood by a human) and analyze again if the number of clusters looks adequate. As the explainability of the data was more than 95% for all cases with just the first two or three principal components, this allowed a good visual representation of the clustering results. Figure 36 and Figure 37 show the 2D plot (two principal components) and the 3D plot (three principal components) of option 3 ($k = 5$):

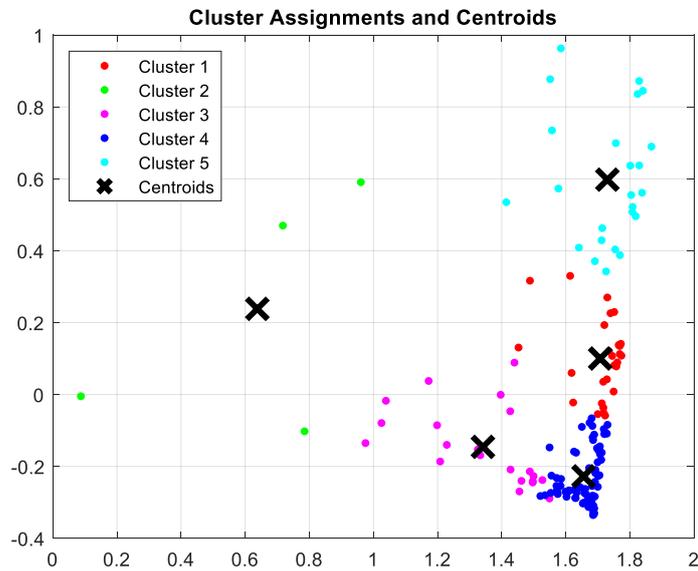


Figure 36. Cluster assignments and centroids in 2D - Option 3 ($k=5$).

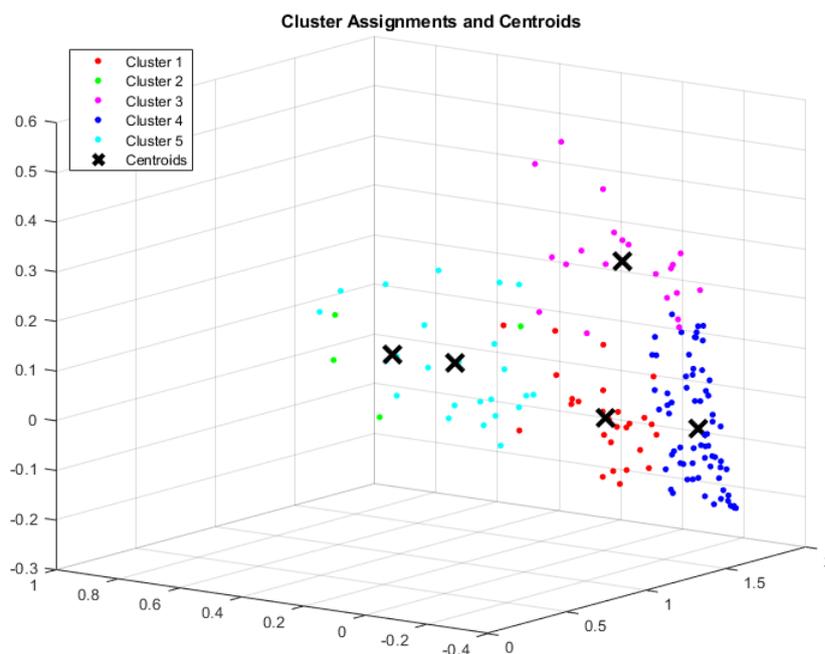


Figure 37. Cluster assignments and centroids in 3D - Option 3 ($k=5$).

- Printing the cluster numbers of each SS into an Excel file using Script 5 and then copy pasting them to the monitoring summary Excel sheet. Later, the dynamic tables that show the average value for each variable and cluster can be analyzed. Table 8 is the example for option 3 ($k = 5$):

Table 8. Cluster results table example - Option 3 ($k = 5$).

Option 3 ($k = 5$)	Nodes		PRIME avail (%)		Short cycle avail (%)		Long cycle avail (%)		Stability histogram										N° SSs
	A.1.3. Avg.	A.1.3. SD	B.1.2. Avg.	B.1.2. SD	A.2.2. Avg.	A.2.2. SD	A.3.2. Avg.	A.3.2. SD	C.1.9. 100- 90%	C.1.9. 90- 80%	C.1.9. 80- 70%	C.1.9. 70- 60%	C.1.9. 60- 50%	C.1.9. 50- 40%	C.1.9. 40- 30%	C.1.9. 30- 20%	C.1.9. 20- 10%	C.1.9. 10- 0%	
1	150	27	98.0%	4.4%	94.1%	9.5%	93.4%	10.4%	24.3%	9.5%	10.2%	13.0%	12.3%	11.4%	9.0%	6.4%	1.9%	2.2%	28
2	36	6	97.4%	5.7%	96.1%	8.3%	94.3%	11.2%	29.8%	8.7%	10.0%	11.2%	12.0%	11.7%	9.2%	4.9%	1.3%	1.2%	69
3	326	55	95.0%	9.1%	88.5%	13.2%	87.5%	14.7%	16.2%	7.6%	9.9%	10.8%	12.5%	15.1%	12.8%	10.2%	4.3%	0.6%	23
4	34	7	83.1%	19.5%	71.6%	28.7%	70.2%	29.8%	10.3%	8.5%	9.0%	11.4%	12.7%	11.9%	15.4%	13.9%	6.5%	0.5%	21
5	91	25	60.8%	9.1%	15.9%	5.1%	20.9%	10.9%	23.4%	6.6%	2.7%	5.0%	5.6%	4.9%	6.4%	20.7%	24.2%	0.5%	4

- Analyzing the formed clusters, trying to understand why the SSs of each cluster are similar and if the clustering made is useful for deployment purposes or not. The individual SSs belonging to each cluster must be checked, as some SSs do not clearly belong to one group and it depends on the particular iteration of the algorithm. This fact happens because k-means starts with the cluster centers (centroids) located at random positions and moves them until the algorithm converges, so they do not always end in the exact same place.

5.6 DEPLOYMENT STRATEGY

The deployment stages will follow the best practices to optimize the knowledge gained and achieve a smooth transition with the first massive PRIME v1.4 smart meter rollout, improving the PRIME v1.3.6 performance as fast as possible.

The main factors that affect the deployment strategy are the following:

- **Equipment arrival:** Most of the PRIME v1.4 smart meters and base nodes have not been received yet and the dates provided by the different manufacturers to supply the equipment are just estimates. The equipment from some manufacturers will arrive some months earlier and it makes sense to start with them in order of arrival.
- **Equipment testing:** Once the equipment is received, testing them in the lab will be necessary in most cases to check if it can be deployed on the field. If nonconformities with the PRIME v1.4 specification are found or the PLC performance in the lab is bad, they must be returned so that the manufacturers fix the problems present.
- **Time constraints:** The objectives are to install PRIME v1.4 on the field in a rapid manner, monitor the SSs to check their performance and achieve good performance as fast as possible.
- **SS specific aspects:** Some concentrators may want to be prioritized, some SS types are easier for installation, proximity between SSs, etc.
- **PRIME v1.3.6 monitoring results:** The SS groups created with the clustering will be used strategically in the different deployment stages with specific objectives in mind.

An additional fact to take into account is that the planning has only been done for type 5 smart meters (smart meter types are explained in 5.2.1), as type 3 and type 4 smart meters will arrive later.

With respect to the PRIME v1.4 manufacturers, there are a total of five, each supplying different types of equipment. The chronogram in Table 9 displays the equipment to be supplied by each manufacturer and the estimated arrival dates. As it can be observed, a few base nodes and smart meters will be available in September and August, but most of the equipment will not arrive until November or December.

In order to have PRIME v1.4 working on the field, all or most type 5 smart meters of the SS must be substituted (three-phase or single-phase) and a base node or two depending on the case (if two transformers are present on the SS). With respect to the type 3 and type 4 meters,

as they are just a few, they will continue working by running PRIME v1.3.6 in parallel with PRIME v1.4 and maintaining the existing PRIME v1.3.6 base node in the SS provisionally.

Table 9. PRIME v1.4 equipment arrival.

Month	August					September				October				November					December				January				
Week	1	2	3	4	5	1	2	3	4	1	2	3	4	1	2	3	4	5	1	2	3	4	1	2	3	4	5
Manufacturer A																											
Type 5 – Single-phase														5,000													
Type 5 – Three-phase						1,000																					
Base Node																			64								
Manufacturer B																											
Base Node																											
Manufacturer C																											
Type 5 – Single-phase										300				4,870													
Type 5 – Three-phase																							1,160				
Manufacturer D																											
Type 5 – Single-phase													450										1,985				
Manufacturer E																											
Type 5 – Single-phase	248																						1,000				
																											1,435

5.7 PRIME v1.4 MONITORING TOOLKIT PREPARATION

This part of the project consists on documenting the work done so that when the deployment takes place, months after finishing this Master Thesis, and the large-scale monitoring is repeated, the know-how is not lost.

The PRIME v1.4 monitoring will be crucial as it will allow the comparison with the PRIME v1.3.6 results, identifying if PLC performance is improved with the latest version of the standard.

The parts to be documented are:

- **PLC performance monitoring:** The different steps and methodology, including the most important aspects of the scripts, the difficulties encountered and the areas where future improvements would be advisable.
- **Data post-processing:** The new additions to the “powerful tool” were documented. In addition, the new Python scripts for data post-processing (the new variable calculation script and the SS data concentration script) and how to use them.
- **SS categorization:** The clustering method applied in the project, the MATLAB script used and how to use it.

Chapter 6. RESULTS DISCUSSION

In this chapter, the main results obtained will be displayed and analysed.

6.1 GRID TOPOLOGY STUDY

6.1.1 ELECTRIC TOPOLOGY STATISTICS

The statistical comparison between the electric topology variables of the three municipalities was performed in a visual manner by means of box plots. The electric variables that were analysed can be separated into five main categories.

6.1.1.1 Customer number and type

When it comes to the number of customers, in Figure 38 it can be appreciated how the median for the three municipalities is very close, and that is one of the reasons why they were chosen in first place. However, the average number of customers is slightly higher in the East municipality.

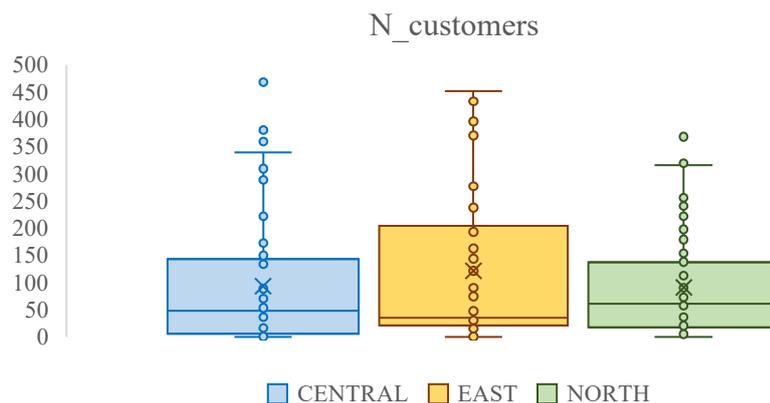


Figure 38. Electrical topology analysis - $N_{customers}$.

It can also be seen how in general there are not many three-phase customers, and the percentage of three-phase customers is notably higher in the Central and East municipalities

(Figure 39), something in principle detrimental for PLC.

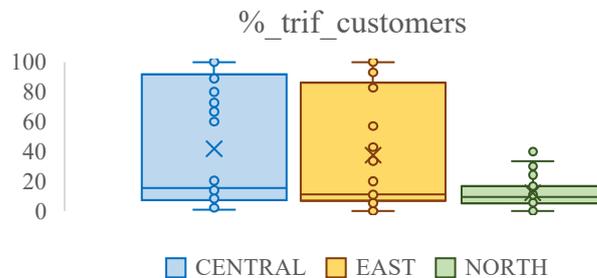


Figure 39. Electrical topology analysis - %_trif_customers.

Finally, when looking at the distribution of customers and their types (Figure 40), the box plots show how there are more non-type 5 customers in the Central and East municipalities, indicating a greater presence of industrial zones. Be that as it may, the number of type 5 customers represents most of the consumption by a great margin.

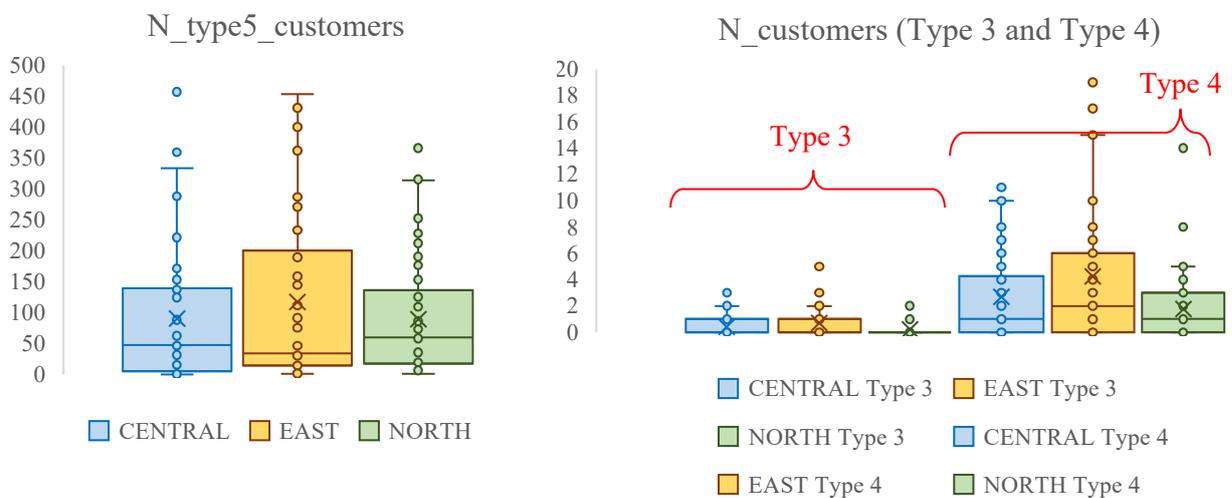


Figure 40. Electrical topology analysis – N_customers per type.

6.1.1.2 Contracted power

The main aspect to be mentioned about the contracted power is that it is greatly correlated with the number of customers and that is why Figure 38 and Figure 41 show exactly the same shape. For that reason, the contracted power does not add extra useful information to the analysis.

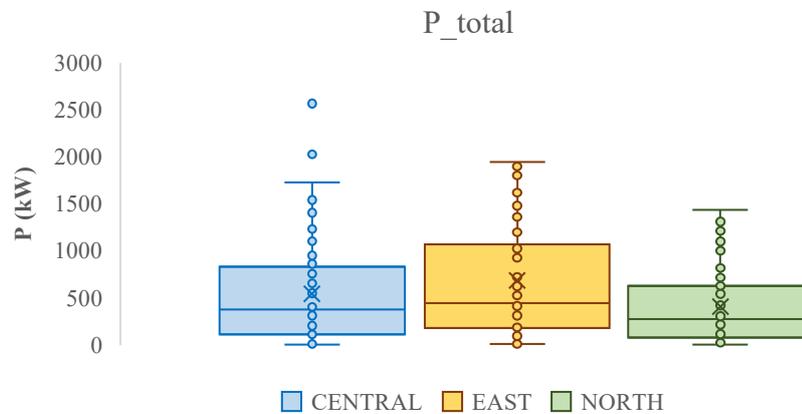


Figure 41. Electrical topology analysis - P_{total} .

6.1.1.3 Feeders

There are many variables to check when it comes to the feeders, so just the most relevant ones will be reviewed.

Firstly, the feeder lengths distinguishing between the overhead (O), underground (U) and total lengths will be reviewed (Figure 42). The plot shows the predominance of overhead feeders in the North municipality and of underground feeders in the other two, something that could give different behaviours to each group. With respect to the total lengths, they are quite similar in the three cases, with a slightly greater value for the East municipality. Longer feeders can suffer from attenuation and have worse PLC performance if enough switches are not present. Secondly, the number of feeders is also larger in the East and Central municipalities (Figure 43), but this does not necessarily imply that the PLC performance is going to be affected by this aspect.

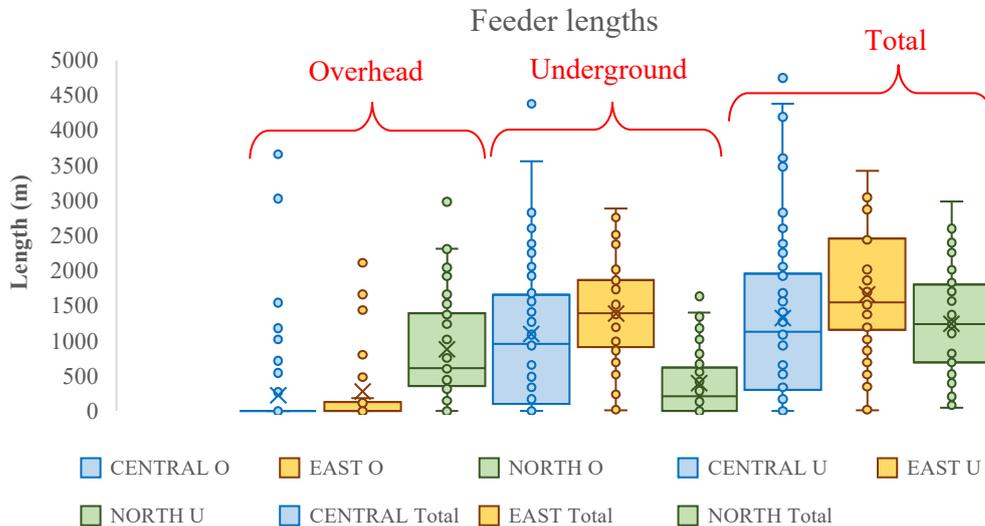


Figure 42. Electrical topology analysis – Feeder lengths.

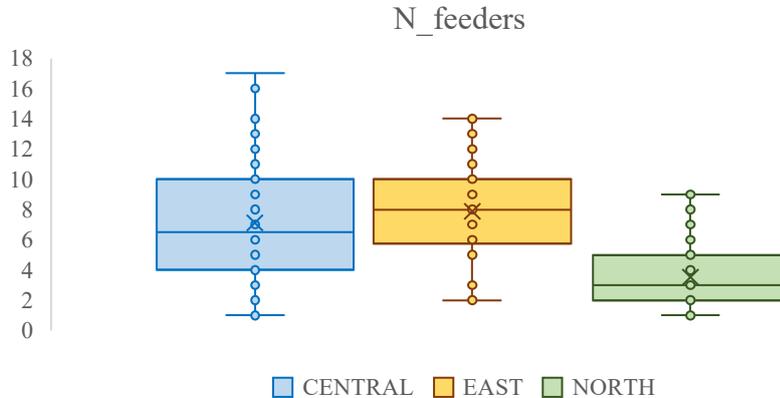


Figure 43. Electric topology analysis - N_feeders.

6.1.1.4 Fuse boxes

The variables that are part of this group are very interesting and can have a great effect in PLC performance. The most relevant ones will be reviewed.

Firstly, the number of fuse boxes per feeder gives an idea of how populated a feeder is, as each fuse box is connected to the smart meters. A more populated feeder can result in more

data collisions but at the same time there are also more repeating opportunities so there is no clear answer to the question of what is better. The main aspects observable in Figure 44 are that the feeders of the North municipality are more populated in general, but the most populated feeders are located in the East municipality.

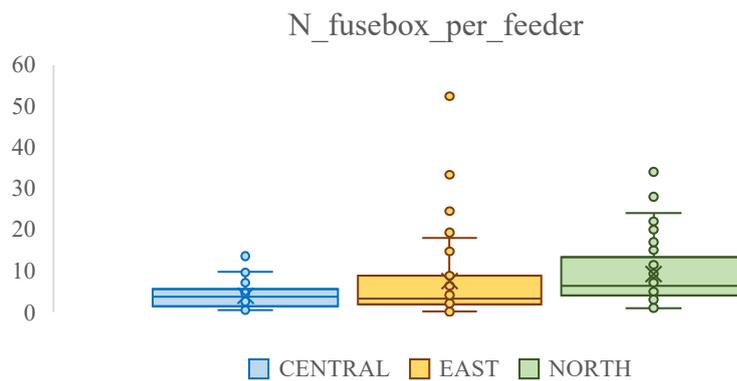


Figure 44. Electric topology analysis - $N_{fusebox_per_feeder}$.

Secondly, the number of customers per fuse box shows how concentrated the meters are in a specific location (Figure 45). This variable has high values when the SS feeds an urban area and the meters are located in meter rooms. A high concentration of meters can damage PLC performance as the medium is shared by all the smart meters and the collisions can increase. The median values for the three municipalities are very similar, having also the three of them atypical SS with very high customer concentrations. The low median values display how this municipalities are not in fact highly populated urban areas.

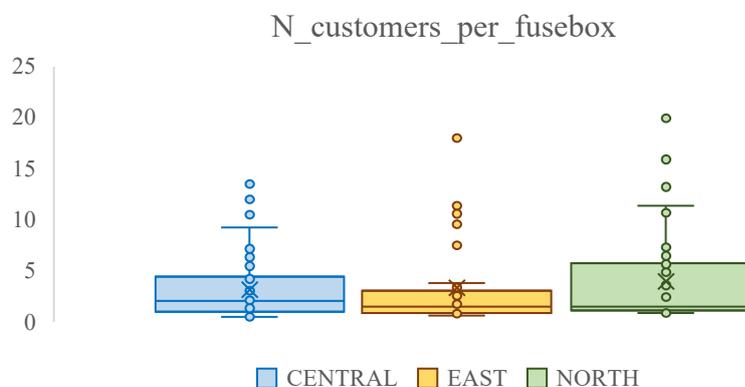


Figure 45. Electrical topology analysis - $N_{customers_per_fusebox}$.

Finally, the length per fuse box represents how spaced are on average the fuse boxes that belong to the same feeder. A higher value of this variable is clearly damaging for PLC as the fuse boxes located at the extremes of the feeders can become isolated as they have no available switches nearby. Figure 46 shows how the medians for the three municipalities are very close and that the fuse boxes can be very disperse sometimes, showing how feeders with very different characteristics are present on each of the three municipalities.

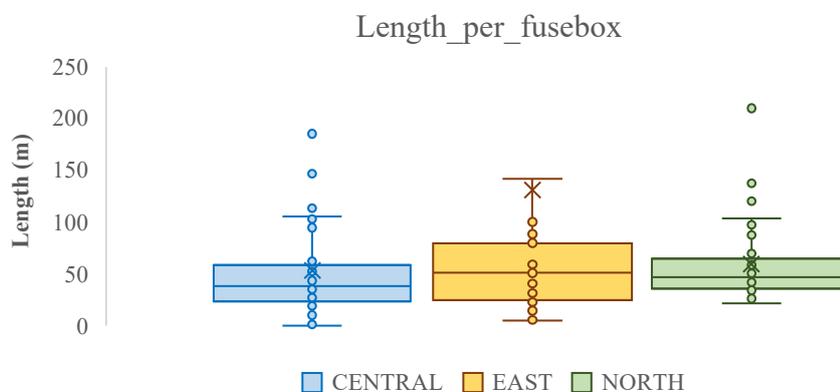


Figure 46. Electric topology analysis - Length_per_fusebox.

6.1.1.5 Load types

With respect to the load types, some loads such as industrial machines can create noises and affect PLC. For that reason, it is important to check the proportions of each type of load present on each municipality (Figure 47).

The three most predominant loads are homes, industrial warehouses and public lighting. The biggest observable difference is the same that was predicted beforehand by looking at the types of customers present, as most loads are from homes and the East and Central municipalities are more industrial than the North counterpart. In fact, the East municipality is the most industrial of them all. With respect to the public lighting, no meaningful differences are observable.

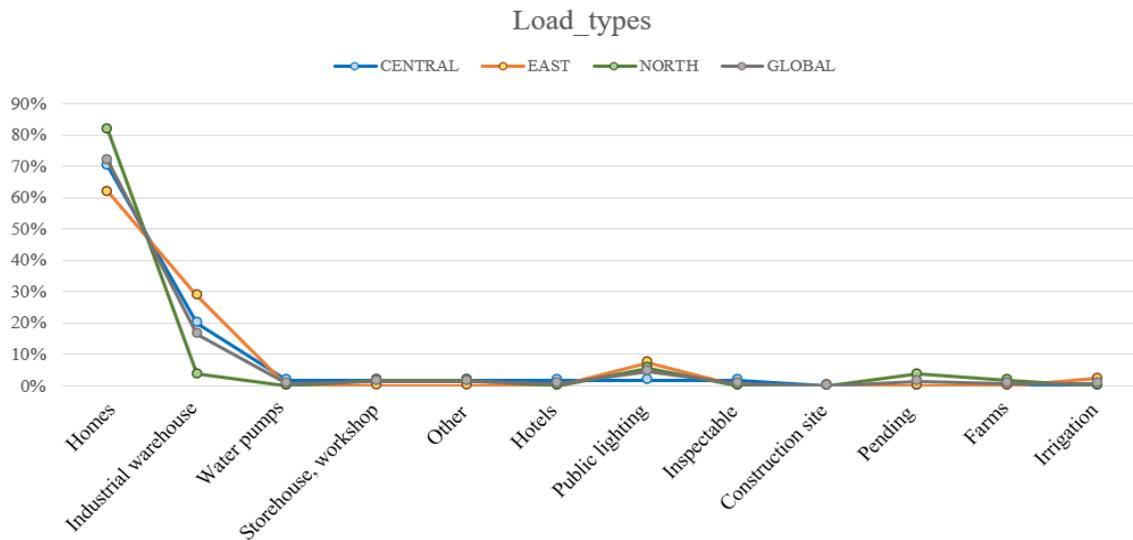


Figure 47. Electrical topology analysis - Load types.

6.1.2 CORRELATION ANALYSIS

The correlation analysis has the aim of finding which of the 26 quantitative electrical variables are highly correlated between them. For that end, after analysing the 26x26 correlation matrix, the following results were obtained (Table 10):

Table 10. Correlated variables.

Group	Correlated variables
1	N_customers, N_type5_customers, P_total, P_type5_customers, N_fusebox_per_feeder, MAX_fusebox_per_feeder.
2	N_trif_customers, N_type4_customers, P_total, P_type4_customers.
3	N_type 3 customers, P_type3_customers.
4	N_gen_customers, P_generation.
5	Feeder_length_O, Max_feeder_length_tot, N_fusebox_per_feeder.
6	Feeder_lengh_U, Feeder_length_tot, N_feeders, N_LVpanel, P_total.
7	N_feeder, N_LVpanel.
8	N_fusebox_per_feeder, MAX_fusebox_per_feeder.

The large number of groups shows how there is a large degree of redundancy present on the variables, and many of them could be removed and still create a good enough draft classification.

Conversely, there are other variables that do not show correlation with any of the other 25, and thus do not add any redundant information: %_trif_customers, N_transf, N_customers_per_fusebox, Length_per_fusebox, MAX_length_per_fusebox, MAX_length_per_fusebox_GENESIS.

6.1.3 SS DRAFT CLASSIFICATION

Taking into account the results obtained with the correlation analysis and after the iterative process explained in 5.2.3, the following three variables were chosen for the SS draft classification:

- **N_customers:** It indicates how many devices share the medium (electric power cable), with its consequent effects in PLC performance. Moreover, the size of the SS is important for deployment considerations, implying how many smart meters need to be substituted.
- **N_customers_per_fusebox:** It indicates how concentrated the smart meters are in specific locations (meter rooms or similar), affecting the probability of data collisions and thus affecting PLC performance.
- **Length_per_fusebox:** It indicates how separated the smart meters that belong to different fuse boxes are on average. If the separation is greater, the probability of finding isolated meters that cannot find a switch is increased, tampering PLC performance. This distinction is only made for SSs with less than 25 customers as when more customers are present the situation is representative of suburban or urban areas and the fuse boxes are in general very close in these cases.

The final results, together with the average and standard deviation (SD) for the only PLC performance variable used on this stage of the project (%_success_S05), which measures the successful daily meter reading percentage, are contained in Table 11:

Table 11. SS draft classification.

		Customers			
		Customers < 25		25 ≤ Customers ≤ 100	Customers > 100
		Length per fuse box ≥ 100	Length per fuse box < 100		
Customers per fuse box	Customers per fuse box < 3	CATEGORY A 20 SSs Average: 89.08% SD: 24.82%	CATEGORY B 41 SSs Average: 97.85% SD: 5.80%	CATEGORY C 33 SSs Average: 97.28% SD: 3.62%	CATEGORY D 15 SSs Average: 98.93% SD: 1.15%
	Customers per fuse box ≥ 3			CATEGORY E 17 SSs Average: 99.47% SD: 0.91%	CATEGORY F 36 SSs Average: 98.6% SD: 2.28%

With respect to the electrical topology, Categories A and B are representative of rural or industrial areas with very few and disperse meters, being the fuse boxes of Category A the ones with the longest distances of separation. Categories C and E are representative of suburban or urban SSs that do not have as many customers. Finally, Categories D and F contain the SSs with at least 100 smart meters, which are normally found in urban areas, where the meters are normally more concentrated (Category F has more SSs than Category E).

With respect to the average daily reading success percentage, Category A shows a lot of variance in the results (high SD), as the SSs with the worst reading percentages are present in this category but the rest have a 100% success rate. The reason for this category having the worst average results is probably that a greater separation between fuse boxes increases the probability of causing smart meter isolation, but this does not always happen even if the fuse boxes are very separated as other external aspects which affect SNR are not reflected in the variables. Category B shows the same behavior as Category A but in a less stressed manner, which indicates that when the length between fuse boxes is reduced the performance is improved on average and also that it is less likely to find a problematic SS. Categories C, D, E and F contain very similar SSs (small standard deviation) and with very good PLC performance. In general, the average performance is improved as the number of customers

is increased (more switching opportunities) but it becomes worse as the concentration of smart meters increases (more chances of collisions happening).

Overall, this first draft classification has not produced useful results to arrange the deployment due to the next reasons:

- There is a clear lack of homogeneity in categories A and B, indicated by the high values of standard deviation, and consequently SSs with excellent or very bad reading success can be found in the same group. This SSs should belong to different groups in order to arrange the deployment with incremental difficulty.
- There is a lot of similarity between categories D, E and F both in terms of the average and standard deviation, indicating that many of the SSs belonging to different categories are in fact very similar and should be in the same one.
- The PLC performance variable used, the average daily reading success, does not provide complete information about how good a SSs behaves. The main reason is the nature of the variable itself, as smart meters readings are tried multiple times a day and if one of those results in a success, the remaining tries are not accounted and a 100% success for that smart meter is registered, which doesn't allow the distinction between smart meters that needed less attempts. Besides, being a single variable that only registers reading success, other very important aspects such as the communications availability or the stability of the SSs are not taken into account.

All in all, this first classification attempt evidences that using exclusively the electric topology to classify the SSs is not enough to form groups with similar PLC behavior, even if it is proven that the electric topology affects it. This happens mainly due to externalities that are not reflected in the electric topology variables (i.e. noisy loads) that reduce SNR and cannot be detected without a detailed analysis of the feeders of each particular SS.

6.2 PLC PERFORMANCE MONITORING

After the completion of the PCL performance monitoring of the 159 SSs (in the end only 145 could be monitored), the overall results by municipality will be reviewed. Additional results considering other aspects can be found in Annex IV: Monitoring Results.

From the whole set of variables that were monitored (Table 7), only the ones that are more relevant and can be analyzed in a global manner will be reviewed.

The overall monitoring results, segregated by municipality, are displayed in Table 12:

Table 12. Monitoring results by municipality (Summary).

Municipality	Nodes		PRIME avail (%)		Short cycle avail (%)		Long cycle avail (%)		Most common topology state							Stability Histogram								
	A.1.3. Avg.	A.1.3. SD	B.1.2. Avg.	B.1.2. SD	A.2.2. Avg.	A.2.2. SD	A.3.2. Avg.	A.3.2. SD	Switches	Switches lv.1	Switches lv.2	Terminals	Switching levels	Lv. most meters	C.1.9. 100-90%	C.1.9. 90-80%	C.1.9. 80-70%	C.1.9. 70-60%	C.1.9. 60-50%	C.1.9. 50-40%	C.1.9. 40-30%	C.1.9. 30-20%	C.1.9. 20-10%	C.1.9. 10-0%
									C.1.2	C.1.3	C.1.4	C.1.5	C.1.6	C.1.7										
Central	104	19	97.0%	6.8%	94.5%	10.8%	93.4%	12.3%	17	5	6	86	3.7	1.8	33%	9%	10%	12%	12%	10%	8%	4%	1%	1%
East	131	19	92.7%	10.1%	86.3%	16.9%	87.0%	18.8%	23	6	7	92	4.9	1.9	14%	7%	10%	11%	12%	14%	14%	11%	5%	2%
North	89	18	92.2%	8.0%	84.6%	10.2%	82.0%	13.2%	13	6	4	72	3.5	1.6	20%	9%	9%	11%	13%	12%	11%	9%	5%	1%
Global	105	18	94.1%	8.1%	88.7%	12.2%	87.5%	14.3%	17	6	6	82	3.9	1.7	24%	8%	10%	11%	12%	12%	11%	8%	3%	1%

The following conclusions can be extracted:

- Both the availabilities (B.1.2, A.2.2 and A.3.2) and the stability (C.1.9) are notably better in the Central municipality. East and North are quite similar, being North a bit more stable and having smaller SD in the availabilities.
- With respect to the number of nodes (A.1.3), obviously, the same results as in Figure 38 are obtained, being East the municipality with a greater average value (more customers).

- The Topic C variables (C.1.2 - C.1.7) do not give any useful insight that can be related to the stability or the availabilities.
- Globally, the monitoring results are quite good, as an average communications availability of 94.1%, an average short cycles availability of 88.7% and an average long cycles availability of 87.5% were achieved. In addition, the overall stability is also good.

6.3 SS CATEGORIZING

In this section, the main insights extracted from the process of categorizing the SSs and a summary of the results obtained are included. For a more detailed analysis of each clustering option that was tried, Annex V: Detailed SS Categorizing can be consulted.

6.3.1 CLUSTERING PROCESS

Firstly, option 1, which only considers the stability histogram was tried. Being it a novel variable, there was some uncertainty but the results it produced were surprisingly good. The four clusters that formed had very homogeneous stabilities (similar shaped curves) and big differences could be observed between clusters, aspects that can be observed in Figure 48. Moreover, a high degree of correlation between the stability and the availabilities was identified, as the average stabilities for each cluster followed the same trend as the availabilities. Besides, correlation between SSs size was also observed, with the most stable clusters having smaller SSs. The drawbacks of this clustering option were that as only stability was considered, some odd SSs with bad availability could be found in the most stable clusters (1 and 2) and these SSs would not be adequate for the first stages of the deployment; and that as SS size was not considered most clusters had mixed sizes and that is not convenient for structuring a smart meter deployment.



Figure 48. Cluster stability curves – Option 1 ($k=4$).

Secondly, option 2 was tried, which only considers the availabilities to perform the clustering. The results were six clusters, two more than before, with SSs with very homogeneous availabilities in each group, no odd SSs being present. The correlation between stability and availability was once again observed, but not to the same extent as before, with the clusters with the best availabilities also having the best stability on average. However, the situation was the opposite to what happened before, some odd SSs with worse stability were present in the clusters with the best availabilities, not being adequate for the first deployment stages. Once again, other disadvantage present was that the SS sizes were also mixed, and this was is not convenient for deployment purposes.

Thirdly, a mix between the availabilities and the SS sizes was tried, producing notable results. The best part of this alternative was that it was an improved version of the previous one, classifying the SSs in very homogeneous groups considering both their size and availabilities. The introduction of the size in the equation proved to be useful, as the groups were more adequate for deployment purposes. Unfortunately, the stability of the SSs, something essential to assess the suitability of a SSs for the first deployment stages was not

considered when forming the clusters, and thus the groups lacked some homogeneity in that aspect, having the possibility of finding SSs with notably different stability on each group.

Fourthly, a combination between stability and availability was tried. The results were by far the best from an academic point of view. If the objective was only to group the SSs according to their PLC performance and no real deployment involved, this would be the choice. The formed groups had superior homogeneity considering both the stability and the availability, and once again displayed the correlation between both variables but leaving the previously mentioned odd SSs (good availability and bad stability or the opposite) in newly formed clusters. The main drawback however was that as SSs size was not considered, the clusters had mixed sizes and it would be inconvenient to structure the deployment. The question as to why in the fourth iteration the SS size was not introduced, as the same problem happened in options 1 and 2, may arise. The reason was that there was an interest as to how these two variables would work combinedly without the intrusion of an additional variable, as the good performance shown by the new histogram caught the interest of Iberdrola.

Finally, all the variables combined were used for the clustering and very good results were achieved. The six formed clusters sacrificed some homogeneity in terms of their stability and availability, but as the size of the SSs was introduced the SSs of each cluster had very similar sizes, which is essential as it was mentioned beforehand.

The summary of the advantages and disadvantages of each of the five clustering options is displayed in Table 13:

Table 13. Clustering option comparison.

Option	Advantages	Disadvantages
1	<ul style="list-style-type: none"> - Very homogeneous stability groups. - Smallest number of clusters. 	<ul style="list-style-type: none"> - Availability not considered, so odd SSs present. - Size not considered, inconvenient for deployment arrangement.

2	- Very homogeneous availability groups.	- Stability not considered, so odd SSs present. - Size not considered, inconvenient for deployment arrangement.
3	- Very convenient groups for arrangement purposes.	- Stability is not considered, so odd SSs are present.
4	- The best from the academic point of view (combination of availability and stability).	- Size is not considered, inconvenient for deployment arrangement.
5	- Best trade-off between size, stability and availability. - Best groups for deployment purposes.	- Size forces cluster membership in some cases, especially for big SSs.

6.3.2 RESULTS

After considering all the pros and cons of each option, option number six was selected because it is clearly the best trade-off between the homogeneity of the groups in terms of availability and stability (PLC performance) and convenience for arranging the deployment. Table 14 provides a summary of the different characteristics of the SSs belonging to each cluster, which will be essential for planning the deployment strategy.

Overall, the results of the clustering were pleasing as the formed clusters will allow to easily structure the deployment strategy, ensuring that the SSs of each cluster have similar PLC performance and size.

Finally, it is worth mentioning that one of the variables that was designed during the project, the stability histogram, resulted to be determining to produce the clusters and to assess the stability of the SSs in a novel way that it had not been done before. Besides, the high degree of correlation found between these variables and the availabilities, traditionally used by Iberdrola, was found. For those reasons, the intention is to continue with the research of this

variable in the future, adding hundreds of new SSs to the analysis to see if the conclusions from the project can be generalised.

Table 14. Cluster characteristics - Selected option.

Cluster number	SS size	Availability	Stability	N° SS
1	Small Very Small	Excellent	Excellent	23
2	Small Medium	Excellent	Very good	39
3	Small Very small	Very good	Good	36
4	Big Very big	Very good	Good	29
5	Small Medium	Bad	Bad	14
6	-	Very bad	Very bad	4

6.4 DEPLOYMENT STRATEGY

After considering all the factors explained in 5.6, specially Table 9 and the clustering results, the deployment was structured into the following main stages with incremental complexity:

- 1. Individual manufacturer trials:** The PRIME v1.3.6 type 5 smart meters of each SS are all substituted by smart meters from the same manufacturer. In this way, the performance of each manufacturer can be proven in a separated way and the possible issues that some of them may have identified easily. However, not every manufacturer has three-phase meters, so a few three-phase meters from other manufacturer may be present, but this will not affect much. Only cluster 1 or 2 SSs are used and only a few SSs per smart meter manufacturer are part of this stage.
- 2. First incremental interoperability trial:** The first interoperability trial (smart meters from different manufacturers are mixed) will be done by choosing a very easy

SS of cluster 2 which has more than 100 smart meters. Firstly, some of the meters of each fuse box will be substituted with meters from two different manufacturers, whose individual performance has been proven in the previous stage. Once the interoperable environment shows good performance, the next manufacturer will be added incrementally, and the process will be repeated until the SSs has smart meters from every manufacturer installed and they operate correctly.

- 3. Small and medium SS size interoperability trials:** Once the interoperability of all the manufacturer smart meters is proven, full mixes of smart meters on different proportions will be installed in small and medium sized SSs (clusters 1, 2, 5 and 6). Special attention will be paid to the SSs of clusters 5 and 6, as they are expected to have more issues than the others.
- 4. Big and very big SS size interoperability trials:** The strategy is the same as in the previous stage but with the biggest SSs. The SSs belonging to this stage are all from cluster 4.

With respect to the 14 SSs have not been monitored as explained in 5.2, the only available information is the average daily meter reading success, so their performance somewhat unpredictable and they will be deployed in stage 3 or stage 4 depending on their size.

The process to be repeated on each stage is to install the smart meters and the base node, monitor the SSs, analyze the results looking for possible issues and compare them to the previous results with PRIME v1.3.6, expecting at least an improvement.

Finally, the detailed deployment stages in chronological order, divided into substages, including the manufacturers to be installed, the estimated dates and their description are displayed in Table 15:

Table 15. Detailed deployment stages.

Sub-stage	Estimated dates	SS categories	Description
1.1	Mid November	1 and 2.	<ul style="list-style-type: none"> ▪ Manufacturer E 1-ph. + Manufacturer A 3-ph. + Manufacturer B base node. <p><u>Bottleneck:</u> Manufacturer E 1-ph. (248). <u>Priority:</u> SSs with more than 20 meters, not post type and few non-type 5 meters. <u>Quantity:</u> 3 or 4 SSs of each category.</p>
1.2	Mid October	1 and 2.	<ul style="list-style-type: none"> ▪ Manufacturer C 1-ph. + Manufacturer A 3-ph. + Manufacturer B base node. <p><u>Bottleneck:</u> Manufacturer C 1-ph. (300). <u>Priority:</u> SSs with more than 20 meters, not post type and few non-type 5 meters. <u>Quantity:</u> 2 or 3 SSs of each category.</p>
2.1	Early November	2.	<ul style="list-style-type: none"> ▪ Manufacturer E 1-ph. + Manufacturer C 1-ph. + Manufacturer A 3-ph. + Manufacturer B base node. <p><u>Quantity:</u> Single SSs of category 2 with at least 100 meters and at least 10 smart meters per fuse box. <u>Procedure:</u> Install 2 or 3 smart meters from Manufacturers E and C on each fuse box and leave the rest. All the three-phase meters are substituted by Manufacturer A meters.</p>
1.3	Early November	1 and 2.	<ul style="list-style-type: none"> ▪ Manufacturer D 1-ph. + Manufacturer A 3-ph. + Manufacturer B base node. <p><u>Bottleneck:</u> Manufacturer D 1-ph. (450). <u>Priority:</u> SSs with more than 20 meters, not post type and few non-type 5 meters. <u>Quantity:</u> 2 or 3 SSs of each category.</p>
1.4	Mid November	1 and 2.	<ul style="list-style-type: none"> ▪ Manufacturer A 1-ph. + Manufacturer A 3-ph. + Manufacturer B base node. <p><u>Bottleneck:</u> No. <u>Priority:</u> SSs with more than 20 meters, not post type and few non-type 5 meters. <u>Quantity:</u> 2 or 3 SSs of each category.</p>
2.2	Stage 1.3 success.	2.	<ul style="list-style-type: none"> ▪ Manufacturer D 1-ph. + Manufacturer E 1-ph. + Manufacturer C 1-ph. + Manufacturer A 3-ph. + Manufacturer B base node. <p><u>Quantity:</u> Same SSs as in substage 2.1. <u>Procedure:</u> Install 2 or 3 smart meters from Manufacturer D on each fuse box, substituting old PRIME v1.3.6 meters.</p>

2.3	Stage 1.4 success	1.	<ul style="list-style-type: none"> ▪ Manufacturer A 1-ph. + Manufacturer C 1-ph. + Manufacturer D 1-ph. + Manufacturer E 1-ph. + Manufacturer A -ph. + Manufacturer B base node. <p><u>Quantity:</u> Same SSs as in substage 2.1.</p> <p><u>Procedure:</u> Install 2 or 3 smart meters from Manufacturer A on each fuse box, substituting old PRIME v1.3.6 meters. All the smart meters on each fuse box are substituted after this substage, and meters from every manufacturer are installed.</p>
1.5	Early December	1, 2 or 3.	<ul style="list-style-type: none"> ▪ Any smart meter manufacturer + Manufacturer A base node <p><u>Bottleneck:</u> No.</p> <p><u>Quantity:</u> 5 or 6 SSs.</p> <p><u>Procedure:</u> Select 2 or 3 SSs from stage 1 that have proven good performance and substitute their base node of Manufacturer B for one of Manufacturer A. Select 3 additional SSs and install the base node together with smart meters from different manufacturers.</p>
3	Stage 2 success	2, 3, 5 and 6.	<ul style="list-style-type: none"> ▪ Any smart meter mix + Manufacturer A or B base node. <p><u>Bottleneck:</u> No.</p> <p><u>Quantity:</u> The remaining small or medium sized SSs.</p>
4	Stage 3 success	4.	<ul style="list-style-type: none"> ▪ Any smart meter mix + Manufacturer A or B base node. <p><u>Bottleneck:</u> Manufacturer E single-phase meters arrive mid-January and it may be necessary to leave 3 or 4 SSs uninstalled until they arrive.</p> <p><u>Quantity:</u> The remaining big or very big sized SSs.</p>

Chapter 7. CONCLUSIONS AND FUTURE WORK

The lack of literature and guidelines to structure smart meter deployments in a standardized manner and the heterogeneity of the electric power grid, not two equal grids existing anywhere in the world, produce an uncertain environment for this type of projects. Besides, this is a pioneer project when it comes to the PLC technology deployed, PRIME v1.4, adding extra uncertainty to the equation.

The methodology applied, an adaptation with additional improvements of the approach developed by Iberdrola for its first PRIME deployments, attempts to ensure the smoothest transition from PRIME v1.3.6 to PRIME v1.4 as possible. The main advantage present if compared to past deployments is the fact that PRIME v1.3.6 is already deployed on the field, so the possibility of monitoring the current performance of the SSs and using the results to categorize the SSs is invaluable, allowing to make decisions supported by that information.

With respect to the topological analysis of the electric power grid, the main conclusion that can be extracted from the project is that due to the numerous externalities that affect PLC, which cannot be reflected in the electric variables present in the databases of the utilities, and the heterogeneity of the LV grid, it is very difficult to predict PLC performance accurately by only considering topological aspects of the LV grid. Obviously, it would be extremely convenient to avoid the tedious process of monitoring all the SSs and achieving a classification that only considering electric topology variables could form groups with homogeneous PLC performance. Consequently, for future PRIME v1.4 deployments with thousands of SSs, once the technology is finally mature and less uncertainty is faced, this could be a good practice if a good set of classification variables can be found. It must not be forgotten either that a study of this kind is essential to know the topological aspects of the SSs, which are also very relevant for planning the deployment. For instance, a SS may have adequate performance and size, but the large number of fuse boxes present make the installation procedure tedious and thus not convenient for the first deployment stages.

With respect to the process of monitoring the SSs and the data post-processing, it would be advisable to add an extra degree of automation if possible, as even with the currently available tools it continues to be a tedious process. The innovations added in this part of the project, will probably be really useful once PRIME v1.4 is deployed, as many of the newly designed variables will provide extra insights about how the latest version of the standard is performing considering its new particularities (new frequency bands, SAR size, etc.), and also allow for a more detailed comparison with PRIME 1.3.6 to observe the improvements.

With respect to the categorization of the SSs, the possibility of applying Machine Learning techniques and achieving homogeneous groups of SSs in terms of stability, availability and size that allow to structure the deployment strategy has been proven. This novel approach had not been tried before and has produced successful results, which could even be improved in the future if more clustering alternatives are tried. Moreover, the newly designed variable, the stability histogram, has proven to be a determining variable for reaching these clusters, while also becoming a new reference to measure stability and showing high correlation with the availabilities (communications, short cycles and long cycles) , which have been traditionally used by Iberdrola to measure PLC performance. Consequently, this variable can be considered as the most remarkable innovation of the project, and the utility has shown its interest to continue studying this variable more in depth by means of the data from thousands of monitorizations that were performed during the last decade, examining if the conclusions reached in this project can be generalized.

With respect to the design of the deployment strategy, the decision to structure the deployment in stages with incremental difficulty in combination with the previously designed SS categories, which contained SSs with suitable characteristics for each of them, produced a favorable framework for the task. This strategy, comprising the dates for each stage, the SSs that form it and the devices to be installed in each SS, is the culmination of the project, structuring the first massive PRIME v1.4 pilot and diminishing the uncertainty faced on this initial step towards the Smart Grid.

Chapter 8. BIBLIOGRAPHY

- [1] S. Galli and T. Lys, "Next generation Narrowband (under 500 kHz) Power Line Communications (PLC) standards," in *China Communications*, vol. 12, no. 3, pp. 1-8, Mar. 2015, doi: 10.1109/CC.2015.7084358.
- [2] A. Sendin, I. Peña, and P. Angueira, "Strategies for Power Line Communications Smart Metering Network Deployment," *Energies*, vol. 7, no. 4, pp. 2377–2420, Apr. 2014 .
- [3] M. Nassar, J. Lin, Y. Mortazavi, A. Dabak, I. H. Kim and B. L. Evans, "Local Utility Power Line Communications in the 3–500 kHz Band: Channel Impairments, Noise, and Standards," in *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 116-127, Sept. 2012, doi: 10.1109/MSP.2012.2187038.
- [4] PRIME Alliance. [Online]. Available: <https://www.prime-alliance.org/>. [Accessed 31 May 2021].
- [5] PRIME Alliance, "PowerLine Intelligent Metering Evolution, v1.3.6." [Online]. Available: https://www.prime-alliance.org/wp-content/uploads/2020/04/PRIME-Spec_v1.3.6.pdf. [Accessed 25 May 2021] .
- [6] PRIME Alliance, "PowerLine Intelligent Metering Evolution, v1.4." [Online]. Available: https://www.prime-alliance.org/wp-content/uploads/2020/04/PRIME-Spec_v1.4-20141031.pdf. [Accessed 31 May 2021].
- [7] I. Arechalde, M. Castro, I. García-Borreguero, A. Sendín, I. Urrutia and A. Fernandez, "Performance of PLC communications in frequency bands from 150 kHz to 500 kHz," 2017 IEEE International Symposium on Power Line Communications and its Applications (ISPLC), 2017, pp. 1-5, doi: 10.1109/ISPLC.2017.7897123.
- [8] D. Dzung, I. Berganza and A. Sendin, "Evolution of powerline communications for smart distribution: From ripple control to OFDM," 2011 IEEE International Symposium on Power Line Communications and Its Applications, 2011, pp. 474-478, doi: 10.1109/ISPLC.2011.5764444.
- [9] A. Sendin, J. Simon, I. Urrutia and I. Berganza, "PLC deployment and architecture for Smart Grid applications in Iberdrola," 18th IEEE International Symposium on Power Line Communications and Its Applications, 2014, pp. 173-178, doi: 10.1109/ISPLC.2014.6812374.

- [10] G3-PLC Alliance, "Narrowband orthogonal frequency division multiplexing power line communication transceivers for G3-PLC networks," [Online]. Available: <https://www.itu.int/rec/T-REC-G.9903-201708-I/en>. [Accessed 19 July 2021].
- [11] CENELEC EN 50065-1, "Signalling on low-voltage electrical installations in the frequency range 3 kHz to 148,5 kHz – Part 1: General requirements, frequency bands and electromagnetic disturbances," 2012.
- [12] PRIME Alliance, "PRIME v1.4 White Paper," [Online]. Available: https://www.prime-alliance.org/wp-content/uploads/2020/04/whitePaperPrimeV1p4_final.pdf. [Accessed 19 July 2021].
- [13] A. Sendin, I. H. Kim, S. Bois, A. Munoz and A. Llano, "PRIME v1.4 evolution: A future proof of reality beyond metering," 2014 IEEE International Conference on Smart Grid Communications (SmartGridComm), 2014, pp. 332-337, doi: 10.1109/SmartGridComm.2014.7007668.
- [14] Jarroba, "Expectation-maximization en Python y Scikit-learn, con ejemplos," [Online]. Available: <https://jarroba.com/expectation-maximization-python-scikit-learn-ejemplos/> [Accessed 4 July 2021].
- [15] Datanovia, "Determining The Optimal Number Of Clusters: 3 Must Know Methods," [Online]. Available: <https://www.datanovia.com/en/lessons/determining-the-optimal-number-of-clusters-3-must-know-methods/>. [Accessed 4 July 2021].
- [16] A. Sendin, I. Berganza, A. Arzuaga, A. Pulkkinen and I. H. Kim, "Performance results from 100,000+ PRIME smart meters deployment in Spain," 2012 IEEE Third International Conference on Smart Grid Communications (SmartGridComm), 2012, pp. 145-150, doi: 10.1109/SmartGridComm.2012.6485974.
- [17] A. Sendin, I. Berganza, A. Arzuaga, X. Osorio, I. Urrutia, and P. Angueira, "Enhanced Operation of Electricity Distribution Grids Through Smart Metering PLC Network Monitoring, Analysis and Grid Conditioning," *Energies*, vol. 6, no. 1, pp. 539–556, Jan. 2013.
- [18] I. Fernandez, I. Angulo, A. Arrinda, D. de la Vega, I. Arechalde, N. Uribe-Pérez and T. Arzuaga, "Characterization of the frequency-dependent transmission losses of the grid up to 500 kHz," in the 25th international conference and exhibition on electricity distribution (CIRED 2019), June 2019.
- [19] I. Fernández, D. de la Vega, D. Roggo, R. Stiegler, L. Capponi, I. Angulo, J. Meyer, and A. Arrinda, "Comparison of Measurement Methods of LV Grid Access Impedance in the

- Frequency Range Assigned to Nb-Plc Technologies,” *Electronics*, vol. 8, no. 10, p. 1155, Oct. 2019.
- [20] I. Fernández, A. Arrinda, I. Angulo, D. De La Vega, N. Uribe-Pérez and A. Llano, "Field Trials for the Empirical Characterization of the Low Voltage Grid Access Impedance From 35 kHz to 500 kHz," in *IEEE Access*, vol. 7, pp. 85786-85795, 2019, doi: 10.1109/ACCESS.2019.2924253.
- [21] J. A. Cortes, L. Diez, F. J. Canete and J. Lopez, "Analysis of the periodic impulsive noise asynchronous with the mains in indoor PLC channels," 2009 IEEE International Symposium on Power Line Communications and Its Applications, 2009, pp. 26-30, doi: 10.1109/ISPLC.2009.4913398.
- [22] I. Fernández, D. de la Vega, A. Arrinda, I. Angulo, N. Uribe-Pérez, and A. Llano, “Field Trials for the Characterization of Non-Intentional Emissions at Low-Voltage Grid in the Frequency Range Assigned to NB-PLC Technologies,” *Electronics*, vol. 8, no. 9, p. 1044, Sep. 2019.
- [23] I. Fernandez, M. García, D. de la Veja, A. Arrinda, I. Angulo, T. Arzuaga, A. Fernández, “Characterization of Non Intentional Conducted Emissions Up to 500 kHz in Urban Environment,” *Renewable Energy and Power Quality Journal*, pp. 663-668, 2018, doi: 10.24084/repqj16.425.
- [24] I. Fernandez, N. Uribe-Pérez, I. Eizmendi, I. Angulo, D. de la Vega, A. Arrinda, T. Arzuaga, “Characterization of non-intentional emissions from distributed energy resources up to 500 kHz: A case study in Spain,” in *International Journal of Electrical Power & Energy Systems*, vol. 105, pp. 549-563, 2019.
- [25] D. Chariag, D. Guezgouz, Y. Raingeaud and J. Lebunetel, "Channel modeling and periodic impulsive noise analysis in indoor Power Line," 2011 IEEE International Symposium on Power Line Communications and Its Applications, 2011, pp. 277-282, doi: 10.1109/ISPLC.2011.5764407.
- [26] V. Arias, “NB-PLC Frequency Bands (151-471 kHz) Feasibility study for PRIME v1.4 Protocol,” Master Thesis in MSc. Industrial Engineering & MSc. Smart Grids, Comillas Pontifical University & University of Strathclyde, 2020.
- [27] I. Arechalde, M. Castro, I. García-Borreguero, A. Sendín, I. Urrutia and A. Fernandez, "Performance of PLC communications in frequency bands from 150 kHz to 500 kHz," 2017 IEEE International Symposium on Power Line Communications and its Applications (ISPLC), 2017, pp. 1-5, doi: 10.1109/ISPLC.2017.7897123.

- [28] B. Nikfar and G. Bumiller, "Real-time synchronization and multiband detection for narrowband power line communication," 2017 IEEE International Symposium on Power Line Communications and its Applications (ISPLC), 2017, pp. 1-5, doi: 10.1109/ISPLC.2017.7897127.
- [29] B. Nikfar, G. Bumiller and A. J. H. Vinck, "An adaptive pursuit strategy for dynamic spectrum assignment in narrowband PLC," 2017 IEEE International Symposium on Power Line Communications and its Applications (ISPLC), 2017, pp. 1-6, doi: 10.1109/ISPLC.2017.7897112.
- [30] FileZilla. [Online]. Available: <https://filezilla-project.org/>. [Accessed 26 July 2021].

ANNEX I: SUSTAINABLE DEVELOPMENT GOALS

The alignment with the SDGs is no doubt part of this Master thesis. Both institutions involved, Universidad Pontificia Comillas and Iberdrola, are aware of their importance and favor the achievement of these goals in their decisions.

The following table contains a summary with the SDGs that are directly aligned with the project and others that are indirectly aligned with it:

Table 16. Sustainable Development Goals.

 <p>7 AFFORDABLE AND CLEAN ENERGY</p>	<p>The rollout of smart meters contributes to the integration of more distributed energy resources (normally renewable energy sources) in the distribution grid. Besides, it also fosters energy efficiency thanks to the new smart grid services that will be developed in the future and that will rely on the Automated Metering Infrastructure.</p>
 <p>9 INDUSTRY, INNOVATION AND INFRASTRUCTURE</p>	<p>The new PRIME v1.4 meters are no doubt an innovation that will be essential for the development of the smart grid, a clear improvement on the current distribution grid.</p>
 <p>13 CLIMATE ACTION</p>	<p>Achieving a more digitalized grid, which facilitates the integration of more clean distributed energy resources, will no doubt contribute to the decarbonisation goals.</p>

ANNEX II: DETAILED SS MONITORING

In this Annex, the details about the monitoring of a SS are presented to showcase the tedious nature of the process, which is not just clicking a button and obtaining the final results. Besides, this process must be repeated 159 times, one per SS, if no automatization is involved.

The ten tasks to be performed in the monitoring of a SS are presented in Figure 49:

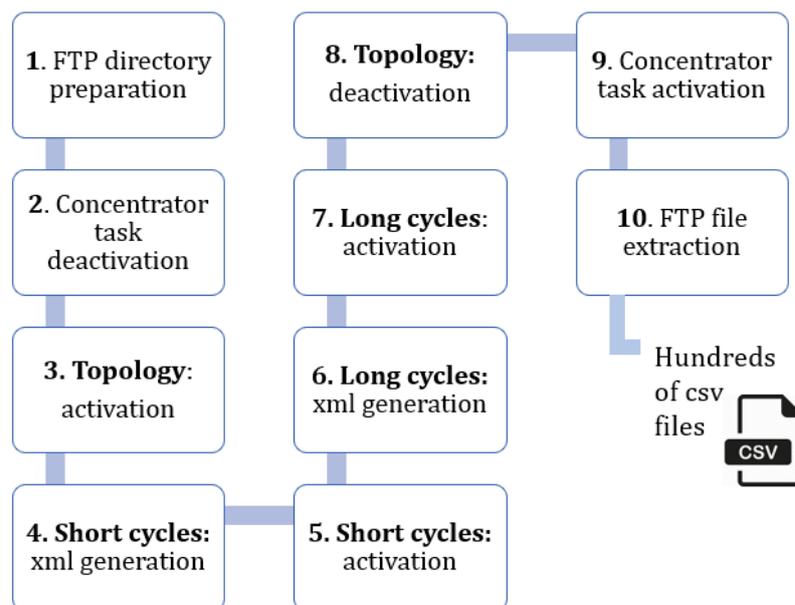
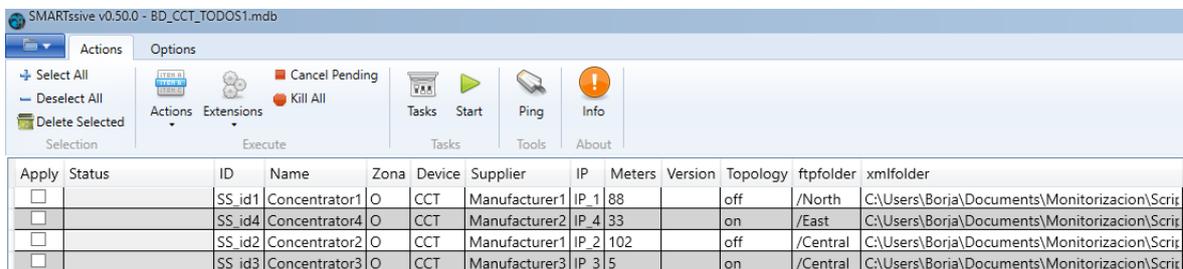


Figure 49. SS monitoring tasks.

The concentrators are accessed via their IP address with a username and password after passing a previous firewall. The main tools used for automatizing the tasks are some Perl scripts and the more user-friendly application “Smartssive”, which was developed by Iberdrola. These tools are used to access the concentrators via scripts, change their settings and activate the different parts of the monitoring that were explained before in an automated way, avoiding the need to perform the 1590 tasks (159 * 10) one by one.

Even if the tools to perform these tasks (Smartssive and Perl scripts) were already developed, the problem was that no massive SS monitoring had been performed since then in Spain and the tools that partially automatized the tasks were outdated and modifications were needed. Therefore, the Python scripts behind Smartssive and the Perl scripts had to be edited and tested multiple times until they worked. These scripts are not part of Annex VI: Scripts because they contain confidential information to access the concentrators and for that reason they will not be displayed in this document. If only a handful of SSs want to be monitored, which is the usual case, this can be done semi-manually without any problem and no modifications would have been needed.

In order to use Smartssive, a Microsoft Access database file with the required attributes of each SS must be filled, which contains the information that the application uses to perform the tasks (Figure 50). Once the database is selected, the desired SSs can be selected on the left side and the desired tasks ordered to be done using the “actions” and “extensions” options.



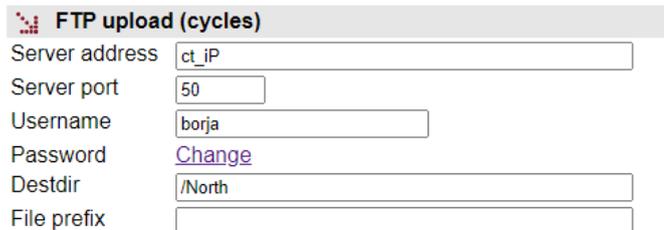
Apply	Status	ID	Name	Zona	Device	Supplier	IP	Meters	Version	Topology	ftpfolder	xmlfolder
<input type="checkbox"/>		SS_id1	Concentrator1	O	CCT	Manufacturer1	IP_1	88		off	/North	C:\Users\Borja\Documents\Monitorizacion\Scrig
<input type="checkbox"/>		SS_id4	Concentrator4	O	CCT	Manufacturer2	IP_4	33		on	/East	C:\Users\Borja\Documents\Monitorizacion\Scrig
<input type="checkbox"/>		SS_id2	Concentrator2	O	CCT	Manufacturer1	IP_2	102		off	/Central	C:\Users\Borja\Documents\Monitorizacion\Scrig
<input type="checkbox"/>		SS_id3	Concentrator3	O	CCT	Manufacturer3	IP_3	5		on	/Central	C:\Users\Borja\Documents\Monitorizacion\Scrig

Figure 50. Smartssive application with example database.

A brief description of each of the ten tasks is provided below for further clarification:

- 1. FTP directory preparation:** The File Transfer Protocol (FTP) FilleZilla [30] was used to upload and store the csv files that the concentrator produces. This task consists on creating a directory in the FTP and later introducing that directory into the settings of the concentrators for uploading the topology and cycles files (Figure 51). The automatization of this task with Smartssive only worked for the topology

files so for the cycle files it had to be done manually as no extra time wanted to be invested trying.



FTP upload (cycles)

Server address

Server port

Username

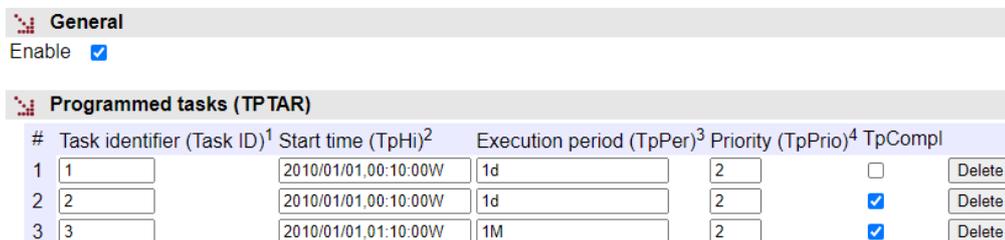
Password [Change](#)

Destdir

File prefix

Figure 51. Cycles FTP directory preparation example.

- 2. Concentrator task deactivation:** As it was mentioned before, the daily tasks that the concentrator does have to be deactivated by unclicking the “enable” option (Figure 52). This task is done automatically with Smartssive.



General

Enable

Programmed tasks (TP TAR)

#	Task identifier (Task ID) ¹	Start time (TpHi) ²	Execution period (TpPer) ³	Priority (TpPrio) ⁴	TpCompl	
1	<input type="text" value="1"/>	<input type="text" value="2010/01/01,00:10:00W"/>	<input type="text" value="1d"/>	<input type="text" value="2"/>	<input type="checkbox"/>	Delete
2	<input type="text" value="2"/>	<input type="text" value="2010/01/01,00:10:00W"/>	<input type="text" value="1d"/>	<input type="text" value="2"/>	<input checked="" type="checkbox"/>	Delete
3	<input type="text" value="3"/>	<input type="text" value="2010/01/01,01:10:00W"/>	<input type="text" value="1M"/>	<input type="text" value="2"/>	<input checked="" type="checkbox"/>	Delete

Figure 52. Concentrator tasks deactivation example.

- 3. Topology activation:** The activation of the topology consists on filling some fields (Figure 53) and selecting the activation. This task was automatized with Smartssive to avoid doing it one by one.

Topology log	
Enable	<input type="checkbox"/>
Filename	<input type="text" value="ct_ID_top"/>
Host	<input type="text" value="IP"/>
Port	<input type="text" value="50"/>
Path	<input type="text" value="/borja"/>
User	<input type="text" value="borja"/>
Pass	<input type="text"/>
Maximum log lines per send	<input type="text" value="2000"/>
Send timeout (seconds)	<input type="text" value="900"/>
Number of retries	<input type="text" value="5"/>
Time between retries (seconds)	<input type="text" value="120"/>

Figure 53. Topology activation example.

4. **Short cycles xml generation:** In order to activate the cycles, a xml file of a certain form must be created. The creation of these xml files (Figure 54) is automatized with Smartssive and it uses the information from the database loaded in the application.

```
<?xml version="1.0"?>
- <cycles>
  - <cycle stop="2021/06/24 08:18" immediate="true" period="1"
    name=" " >
    <get element="2" class="3" obis="1-0:1.8.10.255"/>
    <get element="2" class="7" obis="0-0:21.0.6.255"/>
  </cycle>
</cycles>
```

Figure 54. Short cycle xml file example.

5. **Short cycles activation:** It consists on uploading the previously generated xml file so that the cycles are activated (Figure 55). This task is also automatized with Smartssive.

Concentrator cycles	
<input type="button" value="Return"/>	
Upload the cycles config file of the CCT	
Config file	<input type="button" value="Seleccionar archivo"/> Ningún archivo seleccionado
<input type="button" value="Upload"/>	

Figure 55. Cycles xml upload example.

6. **Long cycles xml generation:** It is similar to task 4 but the xml file is different (Figure 56). It is also automatized with Smartssive.

```
<?xml version="1.0"?>
- <cycles>
  - <cycle stop="2021/06/19 13:12" start="2021/06/17 14:12" period="1"
    name=" " >
    <get element="2" class="8" obis="0-0:1.0.0.255"/>
    <get element="2" class="7" obis="1-0:99.1.0.255" selective_access="structure{structure
      {long_unsigned{8}octet_string{00 00 01 00 00 ff}integer{2}long_unsigned{0}}date_time
      {2021/06/16 00:00:00}date_time{2021/06/16 12:00:00}array{}}"/>
    </cycle>
  </cycles>
```

Figure 56. Long cycle xml file example.

7. **Long cycles activation:** It is exactly the same as the activation of the short cycles but using the corresponding xml files.
8. **Topology deactivation:** It consists on deactivating the topology by unclicking the “enable” option from Figure 53. This task was automatized with Smartssive.
9. **Concentrator task activation:** It is the opposite of task 2 and it is also done with Smartssive.
10. **FTP file extraction:** This task is done manually, and it is basically to download the files from the FTP once the monitoring process is over.

ANNEX III: MONITORING DATA POST-PROCESSING

In this Annex, the details of how the SS raw monitoring data was processed are presented, including both the traditional methodology and the additional steps only included in this project.

For its first PRIME deployments, Iberdrola developed a tool known as “powerful tool”, which processes all the raw monitoring csv files and produces an Excel file per SS with the high-level information that the utility considered useful. It is basically an Excel file with macros that calls a series of scripts and after some time of processing it produces the so called “powerful tool Excel file”.

The vast amount of information present in a powerful tool Excel file requires a learning process in order to be able to use it to its fullest. These files are the cornerstone that allow to understand the full behaviour of a SS and what is happening even at individual smart meter or fuse box level, isolating the PLC problems present and starting the proceedings of looking for a solution. Besides just numerical information, a powerful tool file contains different types of graphs that allow to visualize different information, i.e. the connected nodes during the whole monitoring, in a straight-forward manner.

Figure 57, Figure 58 and Figure 59 show some captures of a powerful excel file, where the confidential information has been hidden, to exemplify how one of these files really is:

SS	Field	Concept	Value	Average	Standard Deviation	Max	Min	Comment
DataBase	SS Name							
DataBase	SS Code							
DataBase	Scenario							
DataBase	DataBase PoS in SS 1							
DataBase	PRIME Meters found in SS 1		155					
DataBase	PRIME Meters NOT found in SS 1		0					
DataBase	NON-PRIME Meters in SS 1							
DataBase	SS PRIME Network							
DataBase	SS PRIME Network found in Database		155					
DataBase	SS PRIME Network found in Database under other SS		1					
DataBase	SS PRIME Network NOT found in Database		1					
Topology	Connected Nodes		157	155.85	30.88	157	0	
Topology	Switches		37.08		4.17	46	0	
Topology	Availability		39.42%		3.17%	100.00%	62.38%	
Topology	Changes in the number of Connected Nodes per Minute		0.16					
Topology	Meters with PRIME availability = 0%		0					
Topology	Meters with PRIME availability < 20%		0					
Topology	Meters with PRIME availability < 85%		1					
Topology	Starting Date		14/06/2021 12:53					
Topology	Ending Date		21/06/2021 07:49					
Short Cycles	Connected Nodes			154.84	2.34	158	137	
Short Cycles	Availability			96.73%	13.64%	100.00%	0.00%	
Short Cycles	Duration			376.63	64.15	1220	808	
Short Cycles	Duration per Meter			6.32				
Short Cycles	Meters with availability 0%			2				
Short Cycles	Meters with availability < 20%			2				
Short Cycles	Starting Date			15/06/2021 05:19				
Short Cycles	Ending Date			17/06/2021 05:26				
Long Cycles	Connected Nodes			154.83	1.24	158	152	
Long Cycles	Availability			96.73%	14.09%	100.00%	0.00%	
Long Cycles	Duration			2366.10	136.64	3151	2664	
Long Cycles	Duration per Meter			15.17				
Long Cycles	Meters with availability 0%			2				
Long Cycles	Meters with availability < 20%			2				
Long Cycles	Starting Date			17/06/2021 11:33				
Long Cycles	Ending Date			19/06/2021 11:22				
Connection	Percentage of Disconnection							

SS	Field	Concept	Value	Comment
Topology	Connected Nodes		134	
Topology	Average Availability		39.64%	
Topology	Meters with PRIME availability = 0%		0	
Topology	Meters with PRIME availability < 20%		0	
Topology	Meters with PRIME availability < 85%		0	
Topology	Starting Date		14/06/2021 12:53	
Topology	Ending Date		21/06/2021 07:49	
Short Cycles	Average Availability		96.46%	
Short Cycles	Meters with availability 0%		1	
Short Cycles	Meters with availability < 20%		1	
Short Cycles	Starting Date		15/06/2021 05:19	
Short Cycles	Ending Date		17/06/2021 05:26	
Long Cycles	Average Availability		98.64%	
Long Cycles	Meters with availability 0%		1	

Figure 57. Powerful tool Excel file example - Summary sheet.

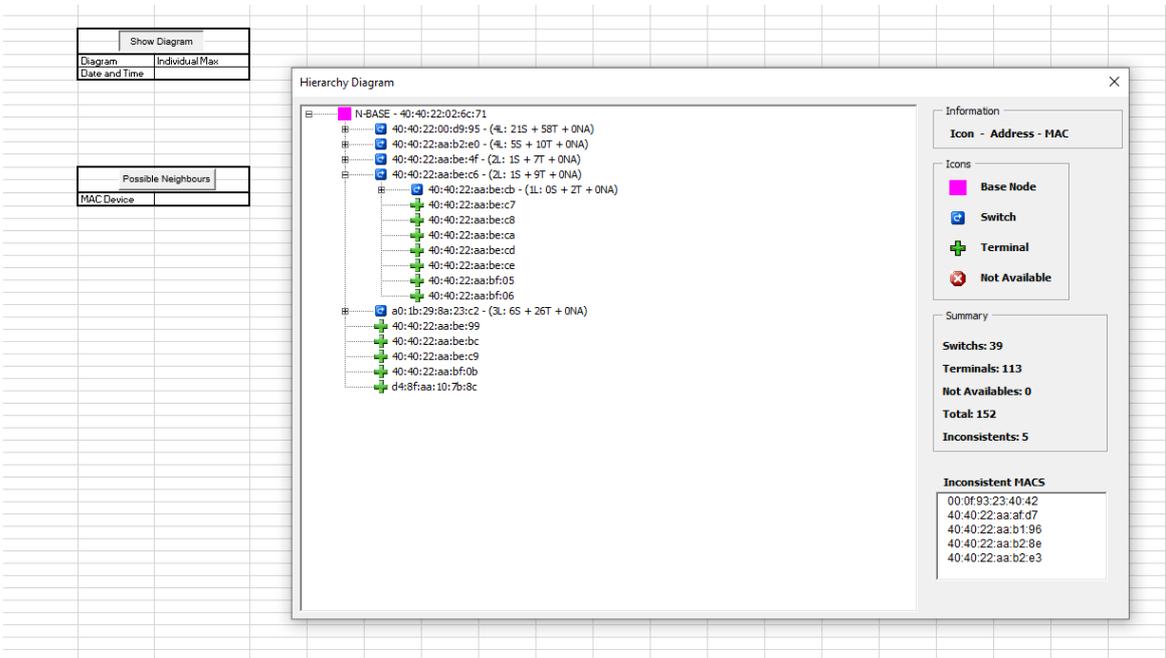


Figure 58. Powerful tool Excel file example - Hierarchy sheet.

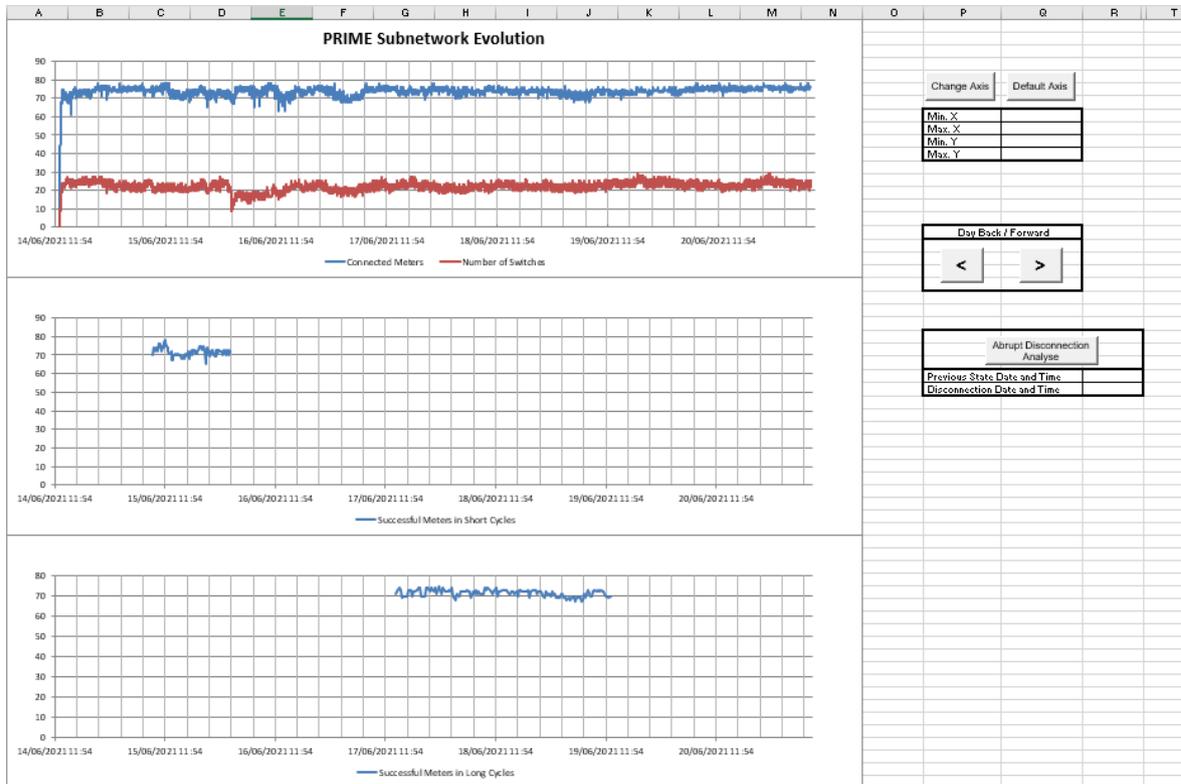


Figure 59. Powerful tool Excel file example - Connected_Nodes_Graph sheet.

Furthermore, in order to introduce the newly designed variables, the macros of the powerful tool had to be modified to create a new sheet in the powerful tool excel files, where that additional information would appear. Figure 60 shows the newly created sheet called “EXTRA_INFORMATION”.

This new sheet is filled by means of a post-processing of the individual powerful tool Excel files with a new Python script that was developed for this end (Script 3). The script uses the information available in the powerful tool Excel files to calculate the newly designed variables and later fill the “EXTRA_INFORMATION” sheet.

A	B	C	D	E	F	G	H	I	J	K	L	M	N
A. Subnetwork Evolution													
Field	Id	Concept	Value	Average	Standard Deviation	Max	Min						
Topology	A.1.2	Periodic Noise	-	-	-	-	-						
Short Cycles	A.2.2	Statistical values of successful meter reading percentage	-	96,17%	1,45%	98,14%	85,09%						
Long Cycles	A.3.2	Statistical values of successful meter reading percentage	-	96,17%	0,77%	98,14%	94,41%						
C. Subnetwork topology													
Field	Id	Concept	Value										
Most common topology state	C.1.2	Number of switches	33										
Most common topology state	C.1.3	Number of switches in level 1	4										
Most common topology state	C.1.4	Number of switches in level 2	16										
Most common topology state	C.1.5	Number of terminals	122										
Most common topology state	C.1.6	Number of switching levels	5										
Most common topology state	C.1.7	Level with the most meters	3										
Most common topology state	C.1.8	Nº of meters with % of time in the most common state	0-10	10-20	20-30	30-40	40-50	50-60	60-70	70-80	80-90	90-100	27
			0	0	0	0	6	24	27	19	25	27	
F. Interoperability													
Field	Id	Concept	Values										
Concentrator information	F.1.1	Name of the manufacturer of the concentrator											
Concentrator information	F.1.2	Firmware version of the concentrator											
Meter information	F.2.1	Names of the manufacturers of the meters											
Meter information	F.2.2	Number of connected meters from each manufacturer	134	9	9	5							
Meter information	F.2.3	Percentage of meters from each manufacturer	85,35%	5,73%	5,73%	3,18%							
Meter availability	F.3.1	Average communications availability for each manufacturer	99,64%	99,76%	95,72%	99,47%							
Meter availability	F.3.2	Average meter availability for each manufacturer short cycle	98,46%	99,69%	77,64%	98,53%							
Meter availability	F.3.3	Average meter availability for each manufacturer long cycle	98,64%	99,47%	75,07%	99,86%							
Topology	F.4.1	Number of switches during most common topology state	28	4	0	1							
Topology	F.4.2	Number of terminals during most common topology state	106	5	9	2							
			1-2	2-3	3-4	4-5	5-6	6-7	7-8	8-9	9-10		
Topology	F.4.3	Switch-terminal manufacturer combinations for each topology level											
G. PRIME tuneable parameters (only for PRIME v1.4)													
Field	Id	Value											
SAR size	G.1.												
Frame length	G.2.												
Modulation robustness management	G.3.												
Beacon frequency	G.4.												
Alive mechanism type	G.5.												
Channels	G.6.												

Figure 60. Powerful tool Excel file example - Extra_Information sheet.

Later, the high-level variables from each SS were concentrated into a single summary Excel file, which would allow to visualize this information for all the monitored SS together and plan the deployment strategy. Besides, this new Excel file allows for the identification of the SSs that perform in a surprising manner and later accessing their corresponding individual powerful tool files to see what is happening in detail. This data concentration was done in an automatized way by means of another Python script (Script 4).

As the whole post-processing can be a bit tricky to understand, a block diagram has been designed to ease the comprehension of everything that explained up to this point (Figure 61):

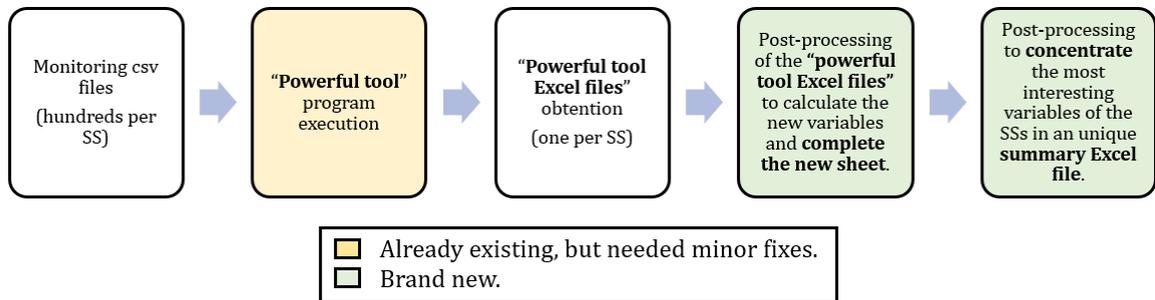


Figure 61. Post-processing block diagram.

ANNEX IV: MONITORING RESULTS

In this Annex, additional results of the monitoring of the 159 SSs are presented and they are also briefly commented.

RESULTS BY CONCENTRATOR MANUFACTURER

Table 17. Monitoring results by concentrator manufacturer (Summary)

Concentrator manufacturer	Nodes		Comms avail (%)		Short avail (%)		Long avail (%)		Most common topology state							Stability Histogram								
	A.1.3. Avg.	A.1.3. SD	B.1.2. Avg.	B.1.2. SD	A.2.2. Avg.	A.2.2. SD	A.3.2. Avg.	A.3.2. SD	Switches	Switches lv.1	Switches lv.2	Terminals	Switching levels	L.v. most meters	C.1.9. 100-90%	C.1.9. 90-80%	C.1.9. 80-70%	C.1.9. 70-60%	C.1.9. 60-50%	C.1.9. 50-40%	C.1.9. 40-30%	C.1.9. 30-20%	C.1.9. 20-10%	C.1.9. 10-0%
									C.1.2	C.1.3	C.1.4	C.1.5	C.1.6	C.1.7										
Manufacturer 1	58	12	88.4%	11.3%	84.3%	13.9%	74.5%	21.3%	9	5	3	47	3.7	1.4	23%	4%	6%	9%	9%	16%	9%	14%	3%	7%
Manufacturer 2	130	37	96.6%	5.7%	88.8%	7.8%	86.4%	8.9%	13	7	4	112	3.3	1.6	38%	5%	8%	10%	13%	13%	%	5%	1%	0%
Manufacturer 3	104	15	94.0%	8.3%	89.1%	12.9%	88.8%	14.8%	19	6	6	79	4.1	1.8	21%	10%	10%	12%	12%	12%	11%	8%	3%	1%
Global	105	18	94.1%	8.1%	88.7%	12.2%	87.5%	14.3%	17	6	6	82	3.9	1.7	24%	9%	10%	11%	12%	12%	11%	8%	2%	1%

In general, the performance displayed by the three different concentrator manufacturers is quite good. The performance of Manufacturer 1 is slightly inferior to the other two, which are quite similar. Besides, while it is true that Manufacturer 2 has the best availability and stability, it must be remarked that only 22 SSs have it installed while 114 SSs have a concentrator from Manufacturer 3 installed. Consequently, as the number of SSs is increased, there are more probabilities of having cases where noises or other externalities are damaging PLC and it is not a fault from the manufacturer. Consequently, Manufacturer 3 is at least equal to Manufacturer 2 in performance terms.

RESULTS BY MUNICIPALITY AND CONCENTRATOR MANUFACTURER

Table 18. Monitoring results by municipality and manufacturer (Summary)

Municipality/ Concentrator manufacturer	Nodes		Comms avail (%)		Short avail (%)		Long avail (%)		Most common topology state							Stability Histogram								
	A.1.3. Avg.	A.1.3. SD	B.1.2. Avg.	B.1.2. SD	A.2.2. Avg.	A.2.2. SD	A.3.2. Avg.	A.3.2. SD	Switches	Switches lv.1	Switches lv.2	Terminals	Switching levels	Lv. most meters	C.1.9.100-90%	C.1.9.90-80%	C.1.9.80-70%	C.1.9.70-60%	C.1.9.60-50%	C.1.9.50-40%	C.1.9.40-30%	C.1.9.30-20%	C.1.9.20-10%	C.1.9.10-0%
									C.1.2	C.1.3	C.1.4	C.1.5	C.1.6	C.1.7										
Central	104	19	97.0%	6.8%	94.5%	10.8%	93.4%	12.3%	17	5	6	86	3.7	1.8	33%	9%	10%	12%	12%	10%	8%	4%	1%	1%
Manufacturer 2	92	27	98.5%	3.5%	97.1%	5.4%	95.5%	7.0%	10	5	4	80	3.1	1.5	48%	4%	7%	9%	12%	10%	6%	4%	0%	0%
Manufacturer 3	108	15	96.4%	8.1%	93.5%	1.9%	92.6%	14.4%	20	5	7	88	4.0	1.8	28%	11%	12%	12%	11%	10%	9%	4%	1%	2%
East	131	19	92.7%	10.1%	86.3%	16.9%	87.0%	18.8%	23	6	7	92	4.9	1.9	14%	7%	10%	11%	12%	14%	14%	11%	5%	2%
Manufacturer 1	74	22	99.2%	4.6%	97.9%	10.9%	93.5%	23.5%	7	5	2	62	3.0	1.0	44%	0%	1%	12%	3%	3%	0%	0%	0%	37%
Manufacturer 3	132	19	92.5%	10.3%	86.0%	17.1%	86.9%	18.6%	24	6	8	93	4.9	1.9	13%	8%	10%	11%	12%	14%	14%	11%	5%	2%
North	89	18	92.2%	8.0%	84.6%	10.2%	82.0%	13.2%	13	6	4	72	3.5	1.6	20%	9%	9%	11%	13%	12%	11%	9%	4%	1%
Manufacturer 1	56	11	87.0%	12.1%	82.6%	14.3%	72.1%	21.0%	9	5	3	46	3.8	1.5	21%	5%	6%	8%	10%	17%	10%	16%	4%	3%
Manufacturer 2	210	58	92.5%	10.2%	70.9%	13.0%	66.9%	13.0%	18	12	4	181	3.6	1.9	17%	6%	12%	13%	14%	18%	9%	9%	2%	0%
Manufacturer 3	74	12	93.2%	6.7%	87.5%	8.9%	86.7%	11.6%	12	5	5	57	3.4	1.5	20%	11%	9%	12%	13%	11%	11%	8%	4%	1%

Differences between the performance of SSs of the same manufacturer on different municipalities can be observed. For instance, Manufacturer 2 shows excellent stability and availability in the Central municipality, but its performance notably decreases in the North municipality. A similar occurrence can be observed if we compare Manufacturer 1 in the East and North municipalities. However, the performance of Manufacturer 3 doesn't vary as much.

The conclusion to be drawn from these facts is that the Central municipality has on average easier environments for PLC, which helps every manufacturer perform consistently, but in the North municipality the environments are the more complex for PLC (more spaced fuse boxes and overhead lines) and the performance of every manufacturer is reduced notably

except for Manufacturer 3. This further supports the idea of the superior consistency of Manufacturer 3.

RESULTS BY LOAD TYPE

Table 19. Monitoring results by load type (Summary)

Load type	Nodes		Comms avail (%)		Short avail (%)		Long avail (%)		Most common topology state							Stability Histogram									
	A.1.3. Avg.	A.1.3. SD	B.1.2. Avg.	B.1.2. SD	A.2.2. Avg.	A.2.2. SD	A.3.2. Avg.	A.3.2. SD	Switches	Switches lv.1	Switches lv.2	Terminals	Switching levels	Lv. most meters	C.1.9.100-90%	C.1.9.90-80%	C.1.9.80-70%	C.1.9.70-60%	C.1.9.60-50%	C.1.9.50-40%	C.1.9.40-30%	C.1.9.30-20%	C.1.9.20-10%	C.1.9.10-0%	
									C.1.2	C.1.3	C.1.4	C.1.5	C.1.6	C.1.7											
Storehouse, workshop	11	1	95.8%	6.5%	93.8%	8.6%	84.3%	15.5%	1		0	10	1.5	1.0	18%	3%	15%	29%	26%	0%	10%	0%	0%	0%	0%
Public lighting	6	1	68.9%	4.2%	67.1%	7.7%	66.8%	8.7%	1	1	0	5	1.1	0.9	22%	3%	7%	7%	15%	7%	7%	3%	0%	0%	
Water pumps	8	2	100%	0%	99.9%	0.1%	99.7%	0.3%	0	0	0	8	1.0	1.0	75%	0%	0%	13%	0%	13%	0%	0%	0%	0%	
Hotels	71	22	98.7%	4.0%	97.9%	8.1%	93.2%	11.4%	8	4	3	65	4.0	2.0	44%	12%	14%	7%	12%	6%	6%	0%	0%	0%	
Industrial warehouse	20	4	70.2%	13.0%	62.6%	19.9%	64.6%	20.0%	8	3	3	14	3.0	1.3	11%	10%	11%	10%	10%	9%	9%	7%	3%	1%	
Other	14	2	94.9%	8.8%	94.1%	6.4%	93.2%	8.2%	3	3	1	12	2.0	1.0	34%	0%	0%	13%	20%	8%	20%	5%	0%	0%	
Irrigation	14	2	96.5%	6.7%	96.7%	3.2%	93.4%	7.5%	6	1	2	8	5.0	3.0	7%	14%	0%	14%	7%	21%	21%	14%	0%	0%	
Homes	128	22	90.9%	6.6%	86.0%	9.9%	84.4%	12.1%	20	7	7	100	4.0	1.7	23%	8%	9%	11%	11%	12%	10%	8%	3%	1%	

The most relevant aspects observed in the results are that home loads (113 SSs) are more suitable for PLC, indicated by the better stability and availabilities compared to the other two most relevant predominant load types, industrial warehouses (26 SSs) and public lighting (7 SSs). This is not surprising at all, as these types of loads introduce more noise in the grid as it was mentioned in section 3.2.2.

With respect to the remaining types of loads, no conclusions can be extracted as there are only 1 or 2 SSs per type and the results are not meaningful due to the lack of data. An additional monitoring of more SSs with these types of predominant loads would be needed.

ANNEX V: DETAILED SS CATEGORIZING

In this annex, the detailed analysis of each of the results for five clustering options tried is explained.

CLUSTERING OPTION 1

The only variable used as input for the clustering was C.1.9.

Table 20. SS categorizing average results per cluster - Option 1 (k=4).

Option 1 (k=4)	Nodes		PRIME avail (%)		Short cycle avail (%)		Long cycle avail (%)		Stability histogram										N° SS
	A.1.3. Avg.	A.1.3. SD	B.1.2. Avg.	B.1.2. SD	A.2.2. Avg.	A.2.2. SD	A.3.2. Avg.	A.3.2. SD	C.1.9. 100- 90%	C.1.9. 90- 80%	C.1.9. 80- 70%	C.1.9. 70- 60%	C.1.9. 60- 50%	C.1.9. 50- 40%	C.1.9. 40- 30%	C.1.9. 30- 20%	C.1.9. 20- 10%	C.1.9. 10- 0%	
1	12	3	99.3%	2.2%	99.8%	0.4%	98.1%	4.6%	87.2%	0.5%	2.6%	4.0%	2.6%	2.5%	0.5%	0.0%	0.0%	0.0%	8
2	77	15	98.2%	5.0%	97.3%	7.2%	95.6%	10.5%	40.0%	10.7%	10.7%	9.8%	10.9%	8.9%	4.4%	1.5%	0.1%	3.0%	44
3	129	22	95.0%	8.9%	90.2%	14.4%	87.8%	16.0%	12.9%	10.2%	11.5%	15.6%	15.0%	14.7%	11.8%	6.3%	1.9%	0.2%	56
4	122	19	89.2%	11.6%	76.6%	17.0%	77.9%	18.4%	6.5%	4.9%	7.2%	8.5%	11.1%	13.8%	18.2%	19.3%	9.8%	0.8%	37

The SSs belonging to each cluster have very similar stabilities and the groups make sense. The SSs with excellent stability belonging to cluster 1, the SSs with very good stability to cluster 2, the SSs with average stability to cluster 3 and the SSs with bad stability to cluster 4. Small SSs (less nodes) are the more stable, and that is why the average number of nodes in cluster 1 is only 12. The correlation between stability and availability has also been proven, as the availabilities clearly get worse as stability decreases.

The size of the groups looks adequate, as no very small groups have been formed. Moreover, the small number of groups makes it attractive.

On the negative side, it is somewhat problematic that there are just a few SS exceptions that even if they are stable, they have bad availabilities, which wouldn't make them appropriate for

the first stages of the deployment. Besides, except in cluster 1, the size of the SSs belonging to each cluster is very variable and that is not convenient for arranging the deployment.

CLUSTERING OPTION 2

Only variables A.2.2, A.3.2 and B.1.2 were used as inputs for the clustering:

Table 21. SS categorizing average results per cluster- Option 2 (k=5).

Option 2 (k = 5)	Nodes		PRIME avail (%)		Short cycle avail (%)		Long cycle avail (%)		Stability histogram										N° SS
	A.1.3. Avg.	A.1.3. SD	B.1.2. Avg.	B.1.2. SD	A.2.2. Avg.	A.2.2. SD	A.3.2. Avg.	A.3.2. SD	C.1.9. 100-90%	C.1.9. 90-80%	C.1.9. 80-70%	C.1.9. 70-60%	C.1.9. 60-50%	C.1.9. 50-40%	C.1.9. 40-30%	C.1.9. 30-20%	C.1.9. 20-10%	C.1.9. 10-0%	
1	99	17	98.8%	3.0%	97.5%	5.0%	96.8%	6.2%	31.5%	9.8%	10.2%	11.0%	11.1%	11.2%	8.1%	4.7%	1.2%	1.2%	67
2	117	17	95.8%	9.0%	92.7%	14.2%	90.6%	18.3%	19.9%	7.4%	9.4%	12.1%	13.3%	13.4%	11.9%	8.3%	2.7%	1.7%	44
3	123	25	89.5%	16.0%	77.2%	24.2%	76.7%	25.2%	13.0%	7.4%	11.3%	12.3%	12.1%	12.7%	16.4%	9.6%	4.7%	0.5%	23
4	34	8	72.0%	24.1%	58.8%	32.1%	53.1%	32.9%	5.2%	9.3%	5.3%	10.1%	16.7%	13.4%	9.1%	21.2%	9.6%	0.2%	7
5	91	25	60.8%	9.1%	15.9%	5.1%	20.9%	10.9%	23.4%	6.6%	2.7%	5.0%	5.6%	4.9%	6.4%	20.7%	24.2%	0.5%	4

The clusters formed have very homogeneous availabilities and thus there are no odd SSs from that point of view in any of the clusters. Cluster 1 contains the SSs with excellent availabilities, cluster 2 the SSs with almost excellent availabilities, cluster 3 the SSs with very good availabilities, cluster 4 the SSs with bad availabilities (especially in the cycles) and cluster 5 the worst SSs.

The correlation between stability and availability is once again seen, as cluster 1 contains the most stable SSs from all the groups and the order is respected. Be that as it may, as only the availabilities were considered for the clustering, it is possible to find some odd unstable SSs in the best clusters (1,2 and 3), which could be problematic for the first deployment stages.

On the negative side, once again, there is a lot of mixture of different sizes in most groups, which is not convenient for arranging the deployment. If the availabilities would be the most important aspect this would be the option to choose. Considering the stability, the option is not bad either, but some caution would be needed with some of the SSs belonging to the first

clusters. Perhaps, the biggest drawback is that the first cluster contains most of the SSs as their availabilities are good, but an extra differentiation considering both sizes and stabilities would make it much better.

CLUSTERING OPTION 3

Only variables A.1.3, A.2.2, A.3.2 and B.1.2 were used as inputs for the clustering:

Table 22. SS categorizing average results per cluster - Option 3 (k=6).

Option 3 (k=6)	Nodes		PRIME avail (%)		Short cycle avail (%)		Long cycle avail (%)		Stability histogram										N° SS
	A.1.3. Avg.	A.1.3. SD	B.1.2. Avg.	B.1.2. SD	A.2.2. Avg.	A.2.2. SD	A.3.2. Avg.	A.3.2. SD	C.1.9. 100-90%	C.1.9. 90-80%	C.1.9. 80-70%	C.1.9. 70-60%	C.1.9. 60-50%	C.1.9. 50-40%	C.1.9. 40-30%	C.1.9. 30-20%	C.1.9. 20-10%	C.1.9. 10-0%	
1	38	7	98.7%	3.6%	97.8%	4.9%	97.1%	6.3%	34.6%	9.1%	9.7%	10.5%	10.9%	10.7%	7.3%	4.5%	1.1%	1.6%	48
2	148	26	97.9%	4.8%	94.1%	9.8%	93.4%	10.7%	24.7%	9.3%	10.3%	13.1%	12.3%	11.3%	8.8%	6.2%	1.8%	2.1%	29
3	326	55	95.0%	9.1%	88.5%	13.2%	87.5%	14.7%	16.2%	7.6%	9.9%	10.8%	12.5%	15.1%	12.8%	10.2%	4.3%	0.6%	23
4	25	3	93.4%	11.9%	89.5%	18.9%	85.0%	24.8%	17.4%	8.8%	10.6%	12.6%	13.5%	13.2%	15.5%	6.1%	1.7%	0.7%	29
5	46	11	77.1%	22.5%	62.4%	31.0%	63.6%	29.7%	5.6%	6.9%	7.6%	10.6%	13.7%	12.7%	12.9%	19.8%	10.1%	0.1%	12
6	91	25	60.8%	9.1%	15.9%	5.1%	20.9%	10.9%	23.4%	6.6%	2.7%	5.0%	5.6%	4.9%	6.4%	20.7%	24.2%	0.5%	4

The common characteristics between the SSs of each cluster are:

1. Small SSs with excellent availability.
2. Medium SSs with excellent availability.
3. Big SSs with very good availability.
4. Small SSs with very good availability.
5. SSs with bad availability, they are small but by mere coincidence.
6. The SSs with the worst availabilities.

The correlation between availability and stability is again observed. But the possibility of finding SSs with not excellent availability in the first 4 clusters (the good behaving SSs) is present.

The best feature of this option is that the combination between availabilities and sizes provides an excellent framework for arranging the deployment because four clusters of SSs

with good availabilities and different sizes have been formed and also two additional clusters for the worst SSs from the whole deployment. The only drawback present is that the stability of each SS belonging to the four best clusters is not guaranteed.

CLUSTERING OPTION 4

Only variables A.2.2, A.3.2, B.1.2 and C.1.9 were used as inputs for the clustering:

Table 23. SS categorizing average results per cluster - Option 4 (k=5).

Option 4 (k=5)	Nodes		PRIME avail (%)		Short cycle avail (%)		Long cycle avail (%)		Stability histogram										N° SS
			B.1.2. Avg.	B.1.2. SD	A.2.2. Avg.	A.2.2. SD	A.3.2. Avg.	A.3.2. SD	C.1.9. 100-90%	C.1.9. 90-80%	C.1.9. 80-70%	C.1.9. 70-60%	C.1.9. 60-50%	C.1.9. 50-40%	C.1.9. 40-30%	C.1.9. 30-20%	C.1.9. 20-10%	C.1.9. 10-0%	
1	12	3	99.3%	2.2%	99.8%	0.4%	98.1%	4.6%	87.2%	0.5%	2.6%	4.0%	2.6%	2.5%	0.5%	0.0%	0.0%	0.0%	7
2	80	16	98.4%	4.5%	97.6%	6.7%	95.9%	10.0%	40.0%	10.8%	10.6%	9.8%	10.8%	8.8%	4.5%	1.6%	0.1%	3.0%	44
3	136	20	96.7%	6.8%	93.6%	11.4%	92.5%	13.0%	11.1%	8.8%	10.1%	13.3%	13.7%	15.2%	14.3%	9.9%	3.2%	0.4%	62
4	100	22	84.9%	17.8%	71.8%	26.3%	70.2%	26.8%	9.4%	7.0%	9.9%	12.2%	13.7%	13.3%	14.8%	13.0%	6.1%	0.5%	28
5	91	25	60.8%	9.1%	15.9%	5.1%	20.9%	10.9%	23.4%	6.6%	2.7%	5.0%	5.6%	4.9%	6.4%	20.7%	24.2%	0.5%	4

The common characteristics between the SSs of each cluster are:

1. SSs with excellent stability and availability.
2. SSs with very good stability and excellent availability.
3. SSs with good stability and very good availability.
4. SSs with bad stability and average availability.
5. SSs with bad stability and the worst availability.

It is very similar to Option 1 but having an additional cluster and also considering the availability variables, which denies the possibility of having SSs with excellent stability but bad availability in the best groups.

From an academic point of view, this would be the best choice, as the groups formed have very homogeneous stability and availability, the most important variables for studying the PLC performance. Be that as it may, as this SS categorization is done for arranging a real

field deployment, the size of the SSs is a very important aspect to consider and the main problem present is that the formed clusters are too heterogeneous in that aspect.

CLUSTERING OPTION 5

All the variables were used as inputs for the clustering:

Table 24. SS categorizing average results per cluster - Option 5 (k=6).

Option 5 (k=6)	Nodes		PRIME avail (%)		Short cycle avail (%)		Long cycle avail (%)		Stability histogram										N° SS
	A.1.3. Avg.	A.1.3. SD	B.1.2. Avg.	B.1.2. SD	A.2.2. Avg.	A.2.2. SD	A.3.2. Avg.	A.3.2. SD	C.1.9. 100- 90%	C.1.9. 90- 80%	C.1.9. 80- 70%	C.1.9. 70- 60%	C.1.9. 60- 50%	C.1.9. 50- 40%	C.1.9. 40- 30%	C.1.9. 30- 20%	C.1.9. 20- 10%	C.1.9. 10- 0%	
1	27	6	98.4%	4.2%	98.9%	3.3%	96.1%	9.5%	59.2%	6.3%	9.0%	5.7%	6.4%	6.0%	3.0%	0.9%	0.0%	3.5%	23
2	101	17	98.7%	3.9%	96.9%	7.8%	96.4%	8.9%	27.8%	11.3%	9.7%	12.8%	13.7%	11.1%	7.0%	4.4%	0.8%	1.5%	39
3	25	3	94.9%	9.5%	91.4%	15.6%	88.4%	18.2%	9.6%	9.0%	10.8%	14.5%	13.7%	15.2%	16.2%	7.9%	2.6%	0.3%	36
4	301	51	95.4%	8.7%	88.7%	13.0%	87.4%	14.0%	15.6%	7.2%	10.5%	11.3%	12.3%	15.0%	13.2%	10.3%	4.0%	0.5%	29
5	50	11	78.8%	21.2%	63.5%	30.3%	66.1%	28.6%	5.6%	7.1%	8.0%	10.4%	13.9%	12.2%	14.0%	19.1%	9.4%	0.5%	14
6	91	25	60.8%	9.1%	15.9%	5.1%	20.9%	10.9%	23.4%	6.6%	2.7%	5.0%	5.6%	4.9%	6.4%	20.7%	24.2%	0.5%	4

The common characteristics between the SSs of each cluster are:

1. Small SSs with excellent availability and excellent or very good stability.
2. Medium or small SSs with excellent availability and very good stability.
3. Small SSs with very good availability and good stability.
4. Big SSs with very good availability and good stability.
5. SSs with bad availability and stability, which are medium or small sized by mere coincidence.
6. Once again, the four worst behaving SSs of the whole deployment.

Basically, it is an improved version of option 3, which also takes into account stability for making the groups, which moves some SSs from clusters 1 and 2 to less stable clusters as they were not as stable as the average SS from cluster they previously belonged to. The size of the groups is also very convenient as the SSs are more evenly distributed than in the other options.

Perhaps, the only drawback is that when introducing the size in the equation some SSs that are very stable and available are moved to clusters of their own size (specially with big SSs as there is only one cluster for them) even if they are better of a bit worse than the average of the cluster. If their performance was much worse than this average, they would be moved to clusters 5 or 6 but this was not the case.

ANNEX VI: SCRIPTS

SCRIPT 1: ELECTRIC TOPOLOGY CORRELATIONS SCRIPT

```
import numpy as np
import pandas as pd
from sklearn.preprocessing import MinMaxScaler

#Read the excel file with the data
file_name = "electric_top_analysis.xlsx"
wb = pd.read_excel(file_name)

#Scaling of the data (Min-Max)
scaler = MinMaxScaler()
results_scaled = scaler.fit_transform(wb)
df_scaled = pd.DataFrame(results_scaled, columns=wb.columns)

#Separate the data by municipalities
df_full = df_scaled.loc[:, "CAN_CLIENTES": "POR_EXITO_LECT_30DIAS"]
df_sanagus = df_scaled.loc[0:62, "CAN_CLIENTES": "POR_EXITO_LECT_30DIAS"]
df_rafbun = df_scaled.loc[62:106, "CAN_CLIENTES": "POR_EXITO_LECT_30DIAS"]
df_zalla = df_scaled.loc[106:, "CAN_CLIENTES": "POR_EXITO_LECT_30DIAS"]

#Calculate the correlations of the dataframe
corr_matrix_full = df_full.corr()
corr_matrix_sanagus = df_sanagus.corr()
corr_matrix_rafbun = df_rafbun.corr()
corr_matrix_zalla = df_zalla.corr()

# Generate the output Excel file
filename = "corr100.xlsx"

writer = pd.ExcelWriter(filename)
corr_matrix_full.to_excel(writer, sheet_name="Todos", header=True, index=True)
corr_matrix_sanagus.to_excel(writer, sheet_name="San Agustin", header=True,
index=True)
corr_matrix_rafbun.to_excel(writer, sheet_name="Rafelbunyol", header=True,
index=True)
corr_matrix_zalla.to_excel(writer, sheet_name="Zalla", header=True, index=True)
writer.save()
writer.close()
```

SCRIPT 2: SS DRAFT CLASSIFICATION SCRIPT

```
import numpy as np
import pandas as pd
import xlrd

# Read the electric topology Excel file
filename = "electric_top_analysis.xlsx"
wb = pd.read_excel(filename)
```

```

wb = wb.loc[:, "ID_USUARIO":"POR_EXITO_LECT_30DIAS"]

#####
##### Classification settings #####3#####
#####

# Select the municipality to classify or all of them
municipio = "TODOS"
if municipio != "TODOS":
    wb = wb[wb["NOM_MUNICIPIO"] == municipio]

# Variables for the classification
var1 = "CAN_CLIENTES"
var2 = "CLIENTES_POR_CGP"
var3 = "METROS_POR_CGP"
# Classification ranges
var1_ranges = [25,100]
var2_ranges = [3]
var3_ranges = [100]

# Alternative classification type.
tipo = 1 # Tipo 1 combines groups A and E and groups B and F, adjusting the
remaining categories.

#####
##### Classification #####
#####

#Formation of the groups
if len(var2_ranges) == 2:
    grupo_A = wb[(wb[var1] < var1_ranges[0]) & (wb[var2] < var2_ranges[0]) &
(wb[var3] > var3_ranges[0])]
    grupo_B = wb[(wb[var1] < var1_ranges[0]) & (wb[var2] < var2_ranges[0]) &
(wb[var3] <= var3_ranges[0])]
    grupo_C = wb[(wb[var1] >= var1_ranges[0]) & (wb[var1] <= var1_ranges[1]) &
(wb[var2] < var2_ranges[0])]
    grupo_D = wb[(wb[var1] > var1_ranges[1]) & (wb[var2] < var2_ranges[0])]
    grupo_E = wb[(wb[var1] < var1_ranges[0]) & (wb[var2] >= var2_ranges[0]) &
(wb[var2] <= var2_ranges[1]) & (wb[var3] > var3_ranges[0])]
    grupo_F = wb[(wb[var1] < var1_ranges[0]) & (wb[var2] >= var2_ranges[0]) &
(wb[var2] <= var2_ranges[1]) & (wb[var3] <= var3_ranges[0])]
    grupo_G = wb[(wb[var1] >= var1_ranges[0]) & (wb[var1] <= var1_ranges[1]) &
(wb[var2] >= var2_ranges[0]) & (wb[var2] <= var2_ranges[1])]
    grupo_H = wb[(wb[var1] > var1_ranges[1]) & (wb[var2] >= var2_ranges[0]) &
(wb[var2] <= var2_ranges[1])]
    grupo_I = wb[(wb[var1] < var1_ranges[0]) & (wb[var2] > var2_ranges[1]) &
(wb[var3] > var3_ranges[0])]
    grupo_J = wb[(wb[var1] < var1_ranges[0]) & (wb[var2] > var2_ranges[1]) &
(wb[var3] <= var3_ranges[0])]
    grupo_k = wb[(wb[var1] >= var1_ranges[0]) & (wb[var1] <= var1_ranges[1]) &
(wb[var2] > var2_ranges[1])]
    grupo_L = wb[(wb[var1] > var1_ranges[1]) & (wb[var2] > var2_ranges[1])]

    ct_matrix = np.empty((3,4), dtype=np.int8)
    ct_matrix[0,0] = len(grupo_A); ct_matrix[0,1] = len(grupo_B); ct_matrix[0,2]
= len(grupo_C); ct_matrix[0,3] = len(grupo_D)
    ct_matrix[1,0] = len(grupo_E); ct_matrix[1,1] = len(grupo_F); ct_matrix[1,2]
= len(grupo_G); ct_matrix[1,3] = len(grupo_H)
    ct_matrix[2,0] = len(grupo_I); ct_matrix[2,1] = len(grupo_J); ct_matrix[2,2]
= len(grupo_k); ct_matrix[2,3] = len(grupo_L)

```

```

elif tipo != 1:
    grupo_A = wb[(wb[var1] < var1_ranges[0]) & (wb[var2] < var2_ranges[0]) &
(wb[var3] > var3_ranges[0])]
    grupo_B = wb[(wb[var1] < var1_ranges[0]) & (wb[var2] < var2_ranges[0]) &
(wb[var3] <= var3_ranges[0])]
    grupo_C = wb[(wb[var1] >= var1_ranges[0]) & (wb[var1] <= var1_ranges[1]) &
(wb[var2] < var2_ranges[0])]
    grupo_D = wb[(wb[var1] > var1_ranges[1]) & (wb[var2] < var2_ranges[0])]
    grupo_E = wb[(wb[var1] < var1_ranges[0]) & (wb[var2] >= var2_ranges[0]) &
(wb[var3] > var3_ranges[0])]
    grupo_F = wb[(wb[var1] < var1_ranges[0]) & (wb[var2] >= var2_ranges[0]) &
(wb[var3] <= var3_ranges[0])]
    grupo_G = wb[(wb[var1] >= var1_ranges[0]) & (wb[var1] <= var1_ranges[1]) &
(wb[var2] >= var2_ranges[0])]
    grupo_H = wb[(wb[var1] > var1_ranges[1]) & (wb[var2] >= var2_ranges[0])]

    ct_matrix = np.empty((2,4), dtype=np.int8)
    ct_matrix[0,0] = len(grupo_A); ct_matrix[0,1] = len(grupo_B); ct_matrix[0,2]
= len(grupo_C); ct_matrix[0,3] = len(grupo_D)
    ct_matrix[1,0] = len(grupo_E); ct_matrix[1,1] = len(grupo_F); ct_matrix[1,2]
= len(grupo_G); ct_matrix[1,3] = len(grupo_H)

else:
    grupo_A = wb[(wb[var1] < var1_ranges[0]) & (wb[var3] > var3_ranges[0])]
    grupo_B = wb[(wb[var1] < var1_ranges[0]) & (wb[var3] <= var3_ranges[0])]
    grupo_C = wb[(wb[var1] >= var1_ranges[0]) & (wb[var1] <= var1_ranges[1]) &
(wb[var2] < var2_ranges[0])]
    grupo_D = wb[(wb[var1] > var1_ranges[1]) & (wb[var2] < var2_ranges[0])]
    grupo_E = wb[(wb[var1] >= var1_ranges[0]) & (wb[var1] <= var1_ranges[1]) &
(wb[var2] >= var2_ranges[0])]
    grupo_F = wb[(wb[var1] > var1_ranges[1]) & (wb[var2] >= var2_ranges[0])]

    ct_matrix = np.empty((2,4))
    ct_matrix[0,0] = len(grupo_A); ct_matrix[0,1] = len(grupo_B); ct_matrix[0,2]
= len(grupo_C); ct_matrix[0,3] = len(grupo_D)
    ct_matrix[1,0] = np.nan; ct_matrix[1,1] = np.nan; ct_matrix[1,2] =
len(grupo_E); ct_matrix[1,3] = len(grupo_F)

#Variable to analyse for the formed groups
analysis_var = "POR_EXITO_LECT_S05"

g = []; h = []; i = []; j = []; k = []; l = []

a = grupo_A.loc[:, analysis_var].to_numpy(); b = grupo_B.loc[:,
analysis_var].to_numpy(); c = grupo_C.loc[:, analysis_var].to_numpy()
d = grupo_D.loc[:, analysis_var].to_numpy(); e = grupo_E.loc[:,
analysis_var].to_numpy(); f = grupo_F.loc[:, analysis_var].to_numpy()

if tipo != 1:
    g = grupo_G.loc[:, analysis_var].to_numpy(); h = grupo_H.loc[:,
analysis_var].to_numpy()
    if len(var2_ranges) > 1:
        i = grupo_I.loc[:, analysis_var].to_numpy(); j = grupo_J.loc[:,
analysis_var].to_numpy(); k = grupo_k.loc[:, analysis_var].to_numpy(); l =
grupo_L.loc[:, analysis_var].to_numpy()

print("A : " + str(a)); print("B : " + str(b)); print("C : " + str(c)); print("D
: " + str(d)); print("E : " + str(e)); print("F : " + str(f)); print("G : " +

```

```

str(g)); print("H : " + str(h)); print("I : " + str(i)); print("J : " + str(j));
print("K : " + str(k)); print("L : " + str(l))

#####
##### Output Excel file preparation #####
#####

# Add the category letter
lista_A = ["A"] * len(grupo_A); lista_B = ["B"] * len(grupo_B); lista_C = ["C"] *
len(grupo_C); lista_D = ["D"] * len(grupo_D); lista_E = ["E"] * len(grupo_E);
lista_F = ["F"] * len(grupo_F)
grupo_A ["CATEGORIA"] = lista_A; grupo_B ["CATEGORIA"] = lista_B; grupo_C
["CATEGORIA"] = lista_C; grupo_D ["CATEGORIA"] = lista_D; grupo_E ["CATEGORIA"] =
lista_E; grupo_F ["CATEGORIA"] = lista_F

if tipo != 1:
    lista_G = ["G"] * len(grupo_G); lista_H = ["H"] * len(grupo_H)
    grupo_G ["CATEGORIA"] = lista_G; grupo_H ["CATEGORIA"] = lista_H
    if len(var2_ranges) > 1:
        lista_I = ["I"] * len(grupo_I); lista_J = ["J"] * len(grupo_J); lista_K =
["K"] * len(grupo_k); lista_L = ["L"] * len(grupo_L)
        grupo_I ["CATEGORIA"] = lista_I; grupo_J ["CATEGORIA"] = lista_J; grupo_k
["CATEGORIA"] = lista_K; grupo_L ["CATEGORIA"] = lista_L

# Combine the dataframes
lengths = []
if tipo == 1:
    frames = [grupo_A, grupo_B, grupo_C, grupo_D, grupo_E, grupo_F]
    lengths.append(len(grupo_A)); lengths.append(len(grupo_B));
lengths.append(len(grupo_C)); lengths.append(len(grupo_D));
lengths.append(len(grupo_E)); lengths.append(len(grupo_F))
elif len(var2_ranges) > 1:
    frames = [grupo_A, grupo_B, grupo_C, grupo_D, grupo_E, grupo_F, grupo_G,
grupo_H, grupo_I, grupo_J, grupo_k, grupo_L]
    lengths.append(len(grupo_A)); lengths.append(len(grupo_B));
lengths.append(len(grupo_C)); lengths.append(len(grupo_D));
lengths.append(len(grupo_E)); lengths.append(len(grupo_F)),
lengths.append(len(grupo_G)); lengths.append(len(grupo_H));
lengths.append(len(grupo_I)); lengths.append(len(grupo_J));
lengths.append(len(grupo_k)); lengths.append(len(grupo_L))
else:
    frames = [grupo_A, grupo_B, grupo_C, grupo_D, grupo_E, grupo_F, grupo_G,
grupo_H]
    lengths.append(len(grupo_A)); lengths.append(len(grupo_B));
lengths.append(len(grupo_C)); lengths.append(len(grupo_D));
lengths.append(len(grupo_E)); lengths.append(len(grupo_F)),
lengths.append(len(grupo_G)); lengths.append(len(grupo_H))

df_categorias = pd.concat(frames, join="inner")
df_categorias.sort_index(ascending=True, inplace=True)

lengths_mun = dict({"SAN AGUSTIN DE GUADALIX": [], "RAFELBUNYOL": [], "ZALLA": []})
munis = ["SAN AGUSTIN DE GUADALIX", "RAFELBUNYOL", "ZALLA"]
if municipio == "TODOS":
    for muni in munis:
        for frame in frames:
            lengths_mun[muni].append(len(frame[frame["NOM_MUNICIPIO"] == muni]))

# Variable analysis

```

```

parametros = ["POR_EXITO_LLECT_S05", "POR_EXITO_LLECT_7DIAS",
"POR_EXITO_LLECT_30DIAS"]
indices = ["CATEGORIA A", "CATEGORIA B", "CATEGORIA C", "CATEGORIA D", "CATEGORIA
E", "CATEGORIA F"]
if tipo != 1:
    indices.append("CATEGORIA G"); indices.append("CATEGORIA H")
    if len(var2_ranges) > 1:
        indices.append("CATEGORIA I"); indices.append("CATEGORIA J");
indices.append("CATEGORIA K"); indices.append("CATEGORIA L")

df_analisis_mean = pd.DataFrame(columns=parametros, index=indices);
df_analisis_std = pd.DataFrame(columns=parametros, index=indices)
if municipio == "TODOS":
    df_analisis_mean_sanagus = pd.DataFrame(columns=parametros, index=indices)
    df_analisis_std_sanagus = pd.DataFrame(columns=parametros, index=indices)
    df_analisis_mean_rafbun = pd.DataFrame(columns=parametros, index=indices)
    df_analisis_std_rafbun = pd.DataFrame(columns=parametros, index=indices)
    df_analisis_mean_zalla = pd.DataFrame(columns=parametros, index=indices)
    df_analisis_std_zalla = pd.DataFrame(columns=parametros, index=indices)

for i in parametros:
    for j in indices:
        letra = j.split(" ")[1]
        grupo = df_categorias[df_categorias["CATEGORIA"] == letra]
        valor_medio = np.nanmean(grupo.loc[:,i])
        desv_tip = np.nanstd(grupo.loc[:,i])
        df_analisis_mean.loc[j,i] = valor_medio
        df_analisis_std.loc[j,i] = desv_tip

#Calculate the statistics by groups
if municipio == "TODOS":
    for i in parametros:
        for j in indices:
            letra = j.split(" ")[1]
            grupo = df_categorias[(df_categorias["CATEGORIA"] == letra) &
(df_categorias["NOM_MUNICIPIO"] == "SAN AGUSTIN DE GUADALIX")]
            valor_medio = np.nanmean(grupo.loc[:,i])
            desv_tip = np.nanstd(grupo.loc[:,i])
            df_analisis_mean_sanagus.loc[j,i] = valor_medio
            df_analisis_std_sanagus.loc[j,i] = desv_tip

    for i in parametros:
        for j in indices:
            letra = j.split(" ")[1]
            grupo = df_categorias[(df_categorias["CATEGORIA"] == letra) &
(df_categorias["NOM_MUNICIPIO"] == "RAFELBUNYOL")]
            valor_medio = np.nanmean(grupo.loc[:,i])
            desv_tip = np.nanstd(grupo.loc[:,i])
            df_analisis_mean_rafbun.loc[j,i] = valor_medio
            df_analisis_std_rafbun.loc[j,i] = desv_tip

    for i in parametros:
        for j in indices:
            letra = j.split(" ")[1]
            grupo = df_categorias[(df_categorias["CATEGORIA"] == letra) &
(df_categorias["NOM_MUNICIPIO"] == "ZALLA")]
            valor_medio = np.nanmean(grupo.loc[:,i])
            desv_tip = np.nanstd(grupo.loc[:,i])
            df_analisis_mean_zalla.loc[j,i] = valor_medio
            df_analisis_std_zalla.loc[j,i] = desv_tip

#Give the desired table form to the dataframe

```

```

df_analisis_mean["////"] = ""
df_datos_todos = pd.concat([df_analisis_mean, df_analisis_std],axis=1)
df_datos_todos["CAN CTs"] = lengths
if municipio == "TODOS":
    df_analisis_mean_sanagus["////"] = ""
    df_datos_sanagus = pd.concat([df_analisis_mean_sanagus,
df_analisis_std_sanagus],axis=1)
    df_datos_sanagus["CAN CTs"] = lengths_mun["SAN AGUSTIN DE GUADALIX"]
    df_analisis_mean_rafbun["////"] = ""
    df_datos_rafbun = pd.concat([df_analisis_mean_rafbun,
df_analisis_std_rafbun],axis=1)
    df_datos_rafbun["CAN CTs"] = lengths_mun["RAFELBUNYOL"]
    df_analisis_mean_zalla["////"] = ""
    df_datos_zalla = pd.concat([df_analisis_mean_zalla,
df_analisis_std_zalla],axis=1)
    df_datos_zalla["CAN CTs"] = lengths_mun["ZALLA"]

df_categoria_A = df_categorias[df_categorias["CATEGORIA"] == "A"]
df_categoria_B = df_categorias[df_categorias["CATEGORIA"] == "B"]
df_categoria_C = df_categorias[df_categorias["CATEGORIA"] == "C"]
df_categoria_D = df_categorias[df_categorias["CATEGORIA"] == "D"]
df_categoria_E = df_categorias[df_categorias["CATEGORIA"] == "E"]
df_categoria_F = df_categorias[df_categorias["CATEGORIA"] == "F"]

# Generate an Excel file with the results
filename = "clasificacion_v4.0.xlsx"

writer = pd.ExcelWriter(filename)
df_categorias.to_excel(writer, sheet_name="Datos", header=True, index=False)
df_datos_todos.to_excel(writer, sheet_name="Analisis_todos", header=True,
index=True)
if municipio == "TODOS":
    df_datos_sanagus.to_excel(writer, sheet_name="Analisis_SanAgustin",
header=True, index=True)
    df_datos_rafbun.to_excel(writer, sheet_name="Analisis_Rafelbunyol",
header=True, index=True)
    df_datos_zalla.to_excel(writer, sheet_name="Analisis_Zalla", header=True,
index=True)

df_categoria_A.to_excel(writer, sheet_name="Categoria_A", header=True,
index=False)
df_categoria_B.to_excel(writer, sheet_name="Categoria_B", header=True,
index=False)
df_categoria_C.to_excel(writer, sheet_name="Categoria_C", header=True,
index=False)
df_categoria_D.to_excel(writer, sheet_name="Categoria_D", header=True,
index=False)
df_categoria_E.to_excel(writer, sheet_name="Categoria_E", header=True,
index=False)
df_categoria_F.to_excel(writer, sheet_name="Categoria_F", header=True,
index=False)

writer.save()
writer.close()

```

SCRIPT 3: PLC PERFORMANCE EXTRA VARIABLE SCRIPT

```
import numpy as np
import pandas as pd
import os
import openpyxl as op
from openpyxl.styles.numbers import FORMAT_PERCENTAGE_00

#####
##### Functions #####
#####

def my_create_top_dict(df_max_depend):
    '''
    Creates a dictionary with the switches and terminals present on each level
    of the topology

    Parameters:
        df_max_depend (Pandas Dataframe): Dataframe from the "MAX_DEPEND" sheet
    of a powerful tool Excel file.

    Returns:
        meter_dict (dict): Dictionary with the switches and terminals present at
    each level of the topology
    '''

    #Keep only the acumulated max result
    df_data_acum = df_max_depend[df_max_depend["Tipo de Maximo"] == "Acumulado"]
    df_data_acum.reset_index(inplace=True)

    #Initialize the dictionary
    meter_dict = {"level_1_switches":[], "level_1_terminals":[],
"level_2_switches":[], "level_2_terminals":[], "level_3_switches":[],
"level_3_terminals":[], "level_4_switches":[], "level_4_terminals":[],
"level_5_switches":[], "level_5_terminals":[], "level_6_switches":[],
"level_6_terminals":[], "level_7_switches":[],
"level_7_terminals":[], "level_8_switches":[], "level_8_terminals":[],
"level_9_switches":[], "level_9_terminals":[], "level_10_terminals":[],
"inconsistent_MACs":[]}

    level_names = ["Padre1", "Padre2", "Padre3", "Padre4", "Padre5", "Padre6",
"Padre7", "Padre8", "Padre9"]
    df_data_acum = df_data_acum.fillna(0)

    #Fill the dictionary
    for i in range(0, len(df_data_acum)):
        df_meter = df_data_acum.iloc[i,:] #Lock the row of the dataframe
        #Separate the inconsistent MACs
        if df_meter["Inconsistencia Jerárquica"] == "YES":
            meter_dict["inconsistent_MACs"].append(df_meter["MAC Equipo"])
            continue
        #Check switch or terminal
        if df_meter["Estado"] == "S":
            state = "switches"
        else:
            state = "terminals"
        j = 0
        max_j = 0 #Number of switching levels
        for level in level_names:
```

```

        j = j + 1
        if j > max_j:
            max_j = j
        if df_meter[level] == 0:
            break
        #Save the meter in the dictionary
        meter_mac = df_meter["MAC Equipo"]
        meter_dict["level_" + str(j) + "_" + state].append(meter_mac)

    return meter_dict

def my_topology_level_num(meter_dictionary):
    '''
    Returns the number of PRIME topology levels.

    Parameters:
        meter_dictionary (dict): Dictionary created with the 'my_create_top_dict'
function.

    Returns:
        number_of_levels (int): Number of PRIME topology levels.
    '''

    #Initialize the variables
    number_of_levels = 0
    names = [("level_1_switches", "level_1_terminals",1), ("level_2_switches",
"level_2_terminals",2), ("level_3_switches", "level_3_terminals",3),
("level_4_switches", "level_4_terminals",4), ("level_5_switches",
"level_5_terminals",5), ("level_6_switches", "level_6_terminals",6),
("level_7_switches", "level_7_terminals",7), ("level_8_switches",
"level_8_terminals",8), ("level_9_switches", "level_9_terminals",9)]

    if len(meter_dict["level_10_terminals"]) > 0:
        number_of_levels = 10
    else:
        #Iterate the tuple level by level
        for index, tuple in enumerate(names):
            switches = tuple[0]
            terminals = tuple[1]
            level = tuple[2]
            #Access the dictionary and check
            if (len(meter_dictionary[switches]) > 0) or
(len(meter_dictionary[terminals]) > 0):
                number_of_levels = level
            else:
                break

    return number_of_levels

def my_topology_switches_in_level(meter_dictionary, level):
    '''
    Returns the number of switches in a level of the topology.

    Parameters:
        meter_dictionary (dict): Dictionary created with the 'my_create_top_dict'
function.
        level (int): Level of the topology selected to count the switches. Can
only have
        values in the range 1-9.

    Returns:

```

```

    terminals_in_level (int): Number of switches in the selected level.
    ...

    key = "level_" + str(level) + "_switches"
    switches_in_level = len(meter_dictionary[key])

    return switches_in_level

#Returns the number of terminals in a level of the topology
def my_topology_terminals_in_level(meter_dictionary, level):
    """
    Returns the number of terminals in a level of the topology.

    Parameters:
        meter_dictionary (dict): Dictionary created with the 'my_create_top_dict'
function.
        level (int): Level of the topology selected to count the terminals. Can
only have
        values in the range 1-10.
    Returns:
        terminals_in_level (int): Number of terminals in the selected level.
    """

    key = "level_" + str(level) + "_terminals"
    terminals_in_level = len(meter_dictionary[key])

    return terminals_in_level

def my_topology_most_populated(meter_dictionary):
    """
    Returns the level of the topology with the most meters (switches +
terminals).

    Parameters:
        meter_dictionary (dict): Dictionary created with the 'my_create_top_dict'
function.
    Returns:
        most_populated (int): Level with the most meters.
    """
    max_count = 0 #Max number of meters in a level
    most_populated = 0 #Most populated meter

    #Iterate for every topology level
    for i in range(1,11):
        if i != 10:
            count = my_topology_switches_in_level(meter_dictionary,i) +
my_topology_terminals_in_level(meter_dictionary,i)
        else:
            count = my_topology_terminals_in_level(meter_dictionary,i)
        #Check max
        if count > max_count:
            max_count = count
            most_populated = i

    return most_populated

def my_number_terminals_and_switches(meter_dictionary):
    """
    Returns the number of terminals and switches in a topological state.

```

```

Parameters:
    meter_dictionary (dict): Dictionary created with the 'my_create_top_dict'
function.
Returns:
    terminal_num (int): Number of terminals.
    switch_num (int): Number of switches.
'''
#meter_dictionary: Created with the my_create_top_dict function.

terminal_num = 0
switch_num = 0

names = [("level_1_switches", "level_1_terminals"), ("level_2_switches",
"level_2_terminals"), ("level_3_switches", "level_3_terminals"),
("level_4_switches", "level_4_terminals"), ("level_5_switches",
"level_5_terminals"), ("level_6_switches", "level_6_terminals"),
("level_7_switches", "level_7_terminals"), ("level_8_switches",
"level_8_terminals"), ("level_9_switches", "level_9_terminals")]

for index, tuple in enumerate(names):
    switches = tuple[0]
    terminals = tuple[1]
    terminal_num += len(meter_dictionary[terminals])
    switch_num += len(meter_dictionary[switches])

terminal_num += len(meter_dictionary["level_10_terminals"])

return terminal_num, switch_num

def my_percentage_time_most_common_state(df_depend, df_max_depend):
    '''
    Returns a dictionary with the meters as the keys and the percentage of time
each
has been on its most common topological state.

Parameters:
    df_depend (Pandas Dataframe): Dataframe from the "DEPEND" sheet of a
powerful tool Excel file.
    df_max_depend (Pandas Dataframe): Dataframe from the "MAX_DEPEND" sheet
of a powerful tool Excel file.

Returns:
    percentage_time_dict (dict): Dictionary with the meters as keys and the
percentage each has
been on its most common state as data.
'''
    percentage_time_dict = {}

    #Get an array with the MACs of all the meters
    df_max_depend = df_max_depend[df_max_depend["Tipo de Maximo"] == "Acumulado"]
    df_max_depend = df_max_depend[df_max_depend["Inconsistencia Jerárquica"] ==
"NO"] #If this line is added inconsistent MACS are not classified
meter_numpy_array = df_max_depend.loc[:, "MAC Equipo"].to_numpy()
meter_array = []
for meter in meter_numpy_array:
    meter_array.append(meter)

#Fill the dictionary with the time percentages
for meter in meter_array:

```

```

#Get the total time in every state
df_meter_raw = df_depend[df_depend["MAC Equipo"] == meter]
time_array = df_meter_raw.loc[:, "Tiempo"].to_numpy(dtype=int)
total_time = np.nansum(time_array)
#Get the total time in the most common state
df_meter = df_max_depend[df_max_depend["MAC Equipo"] == meter]
state_time = df_meter.iloc[0,7]
#Calculate the percentage
time_percentage = (state_time/total_time) * 100
#Save it in the dictionary
percentage_time_dict[meter] = time_percentage

return percentage_time_dict

def my_get_manufacturers(df_nodes_states, df_max_depend):
    """
    Returns a dictionary with the meters as keys and the manufacturer of each
    meter as data.

    Parameters:
        df_nodes_states (Pandas Dataframe): Dataframe from the "NODS_STATES"
        sheet of a powerful tool Excel file.
        df_max_depend (Pandas Dataframe): Dataframe from the "MAX_DEPEND" sheet
        of a powerful tool Excel file.

    Returns:
        manufacturer_dict (dict): Dictionary with the meters as keys and the
        meter manufacturer as data.
    """

    manufacturer_dict = {}

    #Get an array with the MACs of all the meters
    df_max_depend = df_max_depend[df_max_depend["Tipo de Maximo"] == "Acumulado"]
    meter_numpy_array = df_max_depend.loc[:, "MAC Equipo"].to_numpy()
    meter_array = []
    for meter in meter_numpy_array:
        meter_array.append(meter)
    #Search the manufacturer
    for meter in meter_array:
        df_meter = df_nodes_states[df_nodes_states["MAC"] == meter]
        manufacturer = df_meter.iloc[0,27]
        #Fill the dictionary
        manufacturer_dict[meter] = manufacturer
    return manufacturer_dict

def my_get_manufacturer_list(manufacturer_dict):
    """
    Returns a list with all the meter manufacturer names present in the SS.

    Parameters:
        manufacturer_dict (dict): Dictionary created with the
        'my_get_manufacturers' function.

    Returns:
        manufacturer_list (list): List with the meter manufacturer names.
    """
    #manufacturer_dict --> Created with the my_get_manufacturers_function

```

```

manufacturer_list = []
for elements in manufacturer_dict.keys():
    if (manufacturer_dict[elements] not in manufacturer_list) and
(manufacturer_dict[elements] is not np.nan):
        man = manufacturer_dict[elements]
        manufacturer_list.append(man)

return manufacturer_list

def my_get_manufacturer_switches_top(meter_dict, manufacturer_dict):
    """
    Returns a dictionary with how many switches from each manufacturer are
    present in the most common topological state.

    Parameters:
        meter_dict (dict): Dictionary created with the 'my_create_top_dict'
function.
        manufacturer_dict (dict): Dictionary created with the
'my_get_manufacturers' function.

    Returns:
        manufacturer_switches_dict (dict): Dictionary with the manufacturers as
keys and the number of switches
        in the most common topological state as data.
    """

manufacturer_switches_dict = {}
#Get the manufacturers from the manufacturer dictionary
manufacturer_array = []
for elements in manufacturer_dict.keys():
    if manufacturer_dict[elements] not in manufacturer_array:
        man = manufacturer_dict[elements]
        manufacturer_array.append(man)

#Add the manufacturers as keys of the new dictionary
for manufacturer in manufacturer_array:
    manufacturer_switches_dict[manufacturer] = 0

#Iterate the meter dictionary dictionary
for i in range(1,10):
    key = "level_" + str(i) + "_switches"
    switches_level = meter_dict[key]
    for meter in switches_level:
        manufacturer = manufacturer_dict[meter]
        manufacturer_switches_dict[manufacturer] += 1

return manufacturer_switches_dict

def my_get_manufacturer_terminals_top(meter_dict, manufacturer_dict):
    """
    Returns a dictionary with how many terminals from each manufacturer are
    present in the most common topological state.

    Parameters:
        meter_dict (dict): Dictionary created with the 'my_create_top_dict'
function.
        manufacturer_dict (dict): Dictionary created with the
'my_get_manufacturers' function.

    Returns:

```

```

    manufacturer_terminals_dict (dict): Dictionary with the manufacturers as
keys and the number of terminals
    in the most common topological state as data.
    '''

manufacturer_terminals_dict = {}
#Get the manufacturers from the manufacturer dictionary
manufacturer_array = []
for elements in manufacturer_dict.keys():
    if manufacturer_dict[elements] not in manufacturer_array:
        man = manufacturer_dict[elements]
        manufacturer_array.append(man)

#Add the manufacturers as keys of the new dictionary
for manufacturer in manufacturer_array:
    manufacturer_terminals_dict[manufacturer] = 0

#Iterate the meter dictionary dictionary
for i in range(1,10):
    key = "level_" + str(i) + "_terminals"
    terminals_level = meter_dict[key]
    for meter in terminals_level:
        manufacturer = manufacturer_dict[meter]
        manufacturer_terminals_dict[manufacturer] += 1

return manufacturer_terminals_dict

def my_manufacturer_percentage(manufacturer_dict):
    '''
    Returns a dictionary with the manufacturers as keys and the percentage of
meters from each
    manufacturer as data..

    Parameters:
        manufacturer_dict (dict): Dictionary created with the
'my_get_manufacturers' function.

    Returns:
        percentage_dict (dict): Dictionary with the manufacturers as keys and the
percentage of
        meters in the most common topological state as data.
    '''

percentage_dict = {}
#Get the manufacturers from the manufacturer dictionary
manufacturer_array = []
for elements in manufacturer_dict.keys():
    if manufacturer_dict[elements] not in manufacturer_array:
        man = manufacturer_dict[elements]
        manufacturer_array.append(man)

#Add the manufacturers as keys of the new dictionary
for manufacturer in manufacturer_array:
    percentage_dict[manufacturer] = 0

#Fill the dictionary
total_meters = 0
for meter in manufacturer_dict.keys():
    manufacturer = manufacturer_dict[meter]
    percentage_dict[manufacturer] += 1

```

```

    total_meters += 1
    for manufacturer in percentage_dict.keys():
        percentage_dict[manufacturer] = (percentage_dict[manufacturer] /
total_meters) * 100

    return percentage_dict

def my_common_state_ranges(percentage_time_dict, ranges):
    """
    Returns a dictionary with the number of meters whose percentage of time in
    the most common state is on each
    of the inputed ranges.

    Parameters:
        percentage_time_dict (dict): Dictionary created with the
'my_manufacturer_percentage' function.
        ranges (list): List with the selected ranges for the clasification,
example: ["0-20", "20-60","60-100"].

    Returns:
        ranges_dict (dict): Dictionary with the ranges as keys and the percentage
of
        on each range as data.
    """
    ranges_dict = {}

    #Initilize the dicitionary
    for range in ranges:
        ranges_dict[range] = 0

    #Fill the dictionary
    for meter in percentage_time_dict.keys():
        for range in ranges:
            low_lim = int(range.split("-")[0])
            up_lim = int(range.split("-")[1])
            if ((percentage_time_dict[meter] >= low_lim) and
(percentaje_time_dict[meter] < up_lim)) or (percentage_time_dict[meter] == 100):
                ranges_dict[range] += 1
                break

    return ranges_dict

def my_get_manufacturer_top_relations(df_max_depend, manufacturer_dict):
    """
    Returns the meter parent-child relationships present in the most common
topology state.

    Parameters:
        manufacturer_dict (dict): Dictionary created with the
'my_get_manufacturers' function.
        df_max_depend (Pandas Dataframe): Dataframe from the "MAX_DEPEND" sheet
of a powerful tool Excel file.

    Returns:
        top_relations_dict (dict): Dictionary with the level dependencies as keys
("1-2", "2-3", ..., "9-10") and
        the manufacturer relationships present as data.
    """

```

```

#Keep only the accumulated values
df_max_depend = df_max_depend[df_max_depend["Tipo de Maximo"] == "Acumulado"]
df_max_depend = df_max_depend[df_max_depend["Inconsistencia Jerárquica"] ==
"NO"]

#Keep only the interesting columns
df_data = df_max_depend.loc[:,["MAC Equipo", "Estado", "Padre1", "Padre2",
"Padre3", "Padre4", "Padre5", "Padre6", "Padre7", "Padre8", "Padre9"]]

#Create the dataframe for the results
columns = ["MAC Equipo", "Padre1", "Padre2", "Padre3", "Padre4", "Padre5",
"Padre6", "Padre7", "Padre8", "Padre9"]
df_results =pd.DataFrame(columns=columns)

#Reindexing the dataframe
df_data.reset_index(inplace=True)
df_data = df_data.loc[:,["MAC Equipo", "Estado", "Padre1", "Padre2",
"Padre3", "Padre4", "Padre5", "Padre6", "Padre7", "Padre8", "Padre9"]]
df_data= df_data.fillna(0)

#Fill the dataframe substituting the MACs with the manufacturer of each meter
for i in range(0,len(df_data)):
    #Lock a row
    df_meter = df_data.iloc[i,:]
    #Do the substitutions
    for value in columns:
        mac = df_meter[value]
        if mac != 0:
            manufacturer = manufacturer_dict[mac]
            df_results.loc[i,value] = manufacturer

df_results = df_results.fillna(0)
#Get the topological relations
top_relations_dict = { "1-2":[], "2-3":[], "3-4":[], "4-5":[], "5-6":[], "6-
7":[], "7-8":[], "8-9":[], "9-10":[]}
columns_2 = ["Padre9", "Padre8", "Padre7", "Padre6", "Padre5", "Padre4",
"Padre3", "Padre2", "Padre1", "MAC Equipo"] #Co

for i in range(0,len(df_results)):
    #Lock a row
    df_meter = df_results.iloc[i,:]
    #Iterate the columns
    level_up = 0
    level_down = 1
    manufacturer_up = 0
    manufacturer_down = 0
    for value in columns_2:
        if value == "Padre9":
            manufacturer_down = df_meter[value]
        elif value == "MAC Equipo":
            manufacturer_up = manufacturer_down
            manufacturer_down = df_meter[value]
        else:
            manufacturer_up = manufacturer_down
            manufacturer_down = df_meter[value]
    if (manufacturer_up != 0) and (manufacturer_down != 0):
        level_up += 1
        level_down += 1
        key = str(level_up) + "-" + str(level_down)
        relation = manufacturer_up + "-" + manufacturer_down

```

```

        if relation not in top_relations_dict[key]:
            top_relations_dict[key].append(relation)

    return top_relations_dict

def my_get_statistics_success_short(df_ev_short):
    """
    Returns the availability statistics for the short cycles.

    Parameters:
        df_ev_short (Pandas Dataframe): Dataframe from the "EV_SHORT" sheet of a
    powerful tool Excel file.

    Returns:
        short_stats_dict (dict): Dictionary with the statistics (Average,
    Standard deviation, Max and Min)
        as keys.
    """

    short_stats_dict = {"Average":0, "Standard_deviation":0, "Max":0, "Min":0}
    correct_measurements = df_ev_short.loc[:, "Contadores SIN
Error"].to_numpy(dtype=int)
    incorrect_measurements = df_ev_short.loc[:, "Contadores CON
Error"].to_numpy(dtype=int)
    total_measurements = correct_measurements + incorrect_measurements
    percentage_measurements = (correct_measurements / total_measurements)

    #Calculate the stats
    short_stats_dict["Average"] = np.nanmean(percentage_measurements)
    short_stats_dict["Standard_deviation"] = np.nanstd(percentage_measurements)
    short_stats_dict["Max"] = np.nanmax(percentage_measurements)
    short_stats_dict["Min"] = np.nanmin(percentage_measurements)

    return short_stats_dict

def my_get_statistics_success_long(df_ev_long):
    """
    Returns the availability statistics for the long cycles.

    Parameters:
        df_ev_long (Pandas Dataframe): Dataframe from the "EV_LONG" sheet of a
    powerful tool Excel file.

    Returns:
        long_stats_dict (dict): Dictionary with the statistics (Average, Standard
    deviation, Max and Min)
        as keys.
    """

    long_stats_dict = {"Average":0, "Standard_deviation":0, "Max":0, "Min":0}
    correct_measurements = df_ev_long.loc[:, "Contadores SIN
Error"].to_numpy(dtype=int)
    incorrect_measurements = df_ev_long.loc[:, "Contadores CON
Error"].to_numpy(dtype=int)
    total_measurements = correct_measurements + incorrect_measurements
    percentage_measurements = (correct_measurements / total_measurements)

    #Calculate the stats
    long_stats_dict["Average"] = np.nanmean(percentage_measurements)
    long_stats_dict["Standard_deviation"] = np.nanstd(percentage_measurements)

```

```

long_stats_dict["Max"] = np.nanmax(percentage_measurements)
long_stats_dict["Min"] = np.nanmin(percentage_measurements)

return long_stats_dict

def get_manufacturer_summary_value(df_summary, manufacturer, field, concept):
    '''
    Returns the availability statistics for topology or cycles and from the
    desired manufacturer.

    Parameters:
        df_summary (Pandas Dataframe): Dataframe from the "SUMMARY" sheet of a
        powerful tool Excel file.
        manufacturer (str): Name of the manufacturer of the meter.
        field (str): "Topology", "Short Cycles" or "Long Cycles".
        concept (str): Concept that wants to be extracted.

    Returns:
        Value[0] (float): Dictionary with the statistics (Average, Standard
        deviation, Max and Min)
        as keys.
    '''

    #Fix the problem with the name of some of the manufacturers
    if manufacturer == "SOG":
        manufacturer = "SOGECAM"
    elif manufacturer == "LGZ":
        manufacturer = "Landis"
    elif manufacturer == "SAG":
        manufacturer = "SAGEM"
    elif manufacturer == "ITE":
        manufacturer = "ITRON"
    elif manufacturer == "ELS":
        manufacturer = "ELSTER"
    elif manufacturer == "ORB":
        manufacturer = "ORBIS"
    elif manufacturer == "CIR":
        manufacturer = "CIRCUTOR"

    #Extract the part of the dataframe with information corresponding to the
    chosen manufacturer
    manufacturer_i = manufacturer + " Meters"
    first_index = df_summary[df_summary["SS"] == manufacturer_i].index[0]
    last_index = first_index + 19
    df_manufacturer = df_summary.iloc[first_index:last_index,:]

    #Extract the data filtering by the field and concept
    df_manu = df_manufacturer[df_manufacturer["Field"] == field]
    df_manu = df_manu[df_manu["Concept"] == concept]
    if concept == "Average Availability":
        value = df_manu["Average"].to_numpy()
    else:
        value = df_manu["Value"].to_numpy()

    return value[0]

def my_get_availabilities_all_meters(df_nodes_states, field):
    '''
    Returns the availability statistics from the desired field.

```

```

Parameters:
    df_nodes_states (Pandas Dataframe): Dataframe from the "NODS_STATES"
sheet of a powerful tool Excel file.
    field (str): "Topology", "Short Cycles" or "Long Cycles".

Returns:
    availabilities_dict (dict): Dictionary with the statistics (Average,
Standard deviation, Max and Min)
    as keys and the availabilities from the desired fields as data.
    ...

availabilities_dict = {"Average":0, "Standard_deviation":0, "Max":0, "Min":0}
#Get the key to filter
if field == "Topology":
    key = "Available"
elif field == "Short Cycles":
    key = "% Availability Short Cycles"
elif field == "Long Cycles":
    key = "% Availability Long Cycles"
#Get the availability array from the dataframe
end = len(df_nodes_states)
avail_array = df_nodes_states.loc[0:end-2, key].to_numpy(dtype=float)
#Calculate the result
availabilities_dict["Average"] = np.nanmean(avail_array)
availabilities_dict["Standard_deviation"] = np.nanstd(avail_array)
availabilities_dict["Max"] = np.nanmax(avail_array)
availabilities_dict["Min"] = np.nanmin(avail_array)

return availabilities_dict

def my_get_availabilities_meters_this_ct(df_nodes_states, meters_this_ct, field,
value):
    ...
    Returns the desired availability statistic taking only into account the
desired meters. Its use is to avoid
    including meters that belong to other SS but are also unwantedly read from
others in the calculation.

Parameters:
    df_nodes_states (Pandas Dataframe): Dataframe from the "NODS_STATES"
sheet of a powerful tool Excel file.
    meters_this_ct (List): List or array with the meters that want to be part
of the availability calculation.
    field (str): "Topology", "Short Cycles" or "Long Cycles".
    value(str): "Average", "Standard Deviation", "Max" or "Min".

Returns:
    availabilities (float): Desired availability result.
    ...

availabilities_list = []
#Get the key to filter
if field == "Topology":
    key = "Available"
elif field == "Short Cycles":
    key = "% Availability Short Cycles"
elif field == "Long Cycles":
    key = "% Availability Long Cycles"
#Filter the meters from the dataframe and extract the availabilities
for meter in meters_this_ct:

```

```

df_meter = df_nodes_states[df_nodes_states["Meter"] == meter]
availabilities_list.append(df_meter.loc[:,key])
#Convert it into a numpy array
avail_array = np.array(availabilities_list)
#Calculate the result
if value == "Average":
    availability = np.mean(avail_array)
elif value == "Standard Deviation":
    availability = np.std(avail_array)
elif value == "Max":
    availability = np.max(avail_array)
elif value == "Min":
    availability = np.min(avail_array)

return availability

#####
##### Powerful tool Excel file processing #####
#####

#Download the STG Excel file

cgp_data = 0 #Variable to choose if the database information is used or not

if cgp_data == 1:
    df_STG = pd.read_excel("meter_data.xlsx")

#Iterate for every powerful tool file
file_directory = os.path.join(os.getcwd(), "Ficheros")

for filename in os.listdir(file_directory):
    power_filename = os.path.join(file_directory, filename)
    print(power_filename)

#Open the Excel file with openpyxl
wb_power = op.load_workbook(power_filename, keep_vba=True, keep_links=True)

#Read the powerful tool excel file with pandas
#Extract the dataframes from the sheets
df_max_depend = pd.read_excel(power_filename, sheet_name="MAX_DEPEND")
df_depend = pd.read_excel(power_filename, sheet_name="DEPEND")
df_nodes_states = pd.read_excel(power_filename, sheet_name="NODS_STATES")
#Check if there are cycles files and if not go to the next file
if ("EV_SHORT" in wb_power.sheetnames) and ("EV_LONG" in
wb_power.sheetnames):
    df_ev_short = pd.read_excel(power_filename, sheet_name="EV_SHORT")
    df_ev_long = pd.read_excel(power_filename, sheet_name="EV_LONG")
    df_summary = pd.read_excel(power_filename, sheet_name="SUMMARY")
else:
    continue

#Calculate the variables

#Get the auxiliary variables
meter_dict = my_create_top_dict(df_max_depend)
manufacturer_dict = my_get_manufacturers(df_nodes_states, df_max_depend)
percentage_time_dict = my_percentage_time_most_common_state(df_depend,
df_max_depend)

#Calculate the final variables

```

```

#A --> Subnetwork evolution
short_stats_dict = my_get_availability_all_meters(df_nodes_states, "Short
Cycles") #A.2.2
long_stats_dict = my_get_availability_all_meters(df_nodes_states, "Long
Cycles") #A.3.2

#C --> Subnetwork topology
number_of_levels = my_topology_level_num(meter_dict) #C.1.6
[num_terminals, num_switches] = my_number_terminals_and_switches(meter_dict)
#C.1.5 and C.1.2
num_switches_lev_1 = my_topology_switches_in_level(meter_dict, 1) #C.1.3
num_switches_lev_2 = my_topology_switches_in_level(meter_dict, 2) #C.1.4
most_populated_level = my_topology_most_populated(meter_dict) #C.1.7
ranges_list = ["0-10", "10-20", "20-30", "30-40", "40-50", "50-60", "60-70",
"70-80", "80-90", "90-100"]
ranges_dict = my_common_state_ranges(percentage_time_dict, ranges_list)
#C.1.8

#F --> Interoperability
manufacturer_concentrator = df_summary.iloc[0,8] #F.1.1
if manufacturer_concentrator == "CURRENT":
    manufacturer_concentrator = "ORM"
manufacturer_list = my_get_manufacturer_list(manufacturer_dict) #F.2.1
manufacturer_percentage_dict = my_manufacturer_percentage(manufacturer_dict)
#F.2.3
manufacturer_switches_dict = my_get_manufacturer_switches_top(meter_dict,
manufacturer_dict) #F.4.1
manufacturer_terminals_dict = my_get_manufacturer_terminals_top(meter_dict,
manufacturer_dict) #F.4.2
top_relations_dict = my_get_manufacturer_top_relations(df_max_depend,
manufacturer_dict) #F.4.3

#Fill the new sheet ("EXTRA_INFORMATION") of the powerful tool Excel file

#Read the "EXTRA_INFORMATION" sheet
#Check if it exists (in the old powerful tool Excel files it doesn't)
if "EXTRA_INFORMATION" in wb_power.sheetnames: #If it exists open it
    wb_extra_info = wb_power["EXTRA_INFORMATION"]
else: #If it doesn't exist, create it
    wb_extra_info = wb_power.create_sheet("EXTRA_INFORMATION")

#Add the info to the "EXTRA_INFORMATION"
#A. Subnetwork evolution
wb_extra_info["F5"].value = short_stats_dict["Average"]
wb_extra_info["G5"].value = short_stats_dict["Standard_deviation"]
wb_extra_info["H5"].value = short_stats_dict["Max"]
wb_extra_info["I5"].value = short_stats_dict["Min"]
wb_extra_info["F6"].value = long_stats_dict["Average"]
wb_extra_info["G6"].value = long_stats_dict["Standard_deviation"]
wb_extra_info["H6"].value = long_stats_dict["Max"]
wb_extra_info["I6"].value = long_stats_dict["Min"]

#C. Subnetwork topology
wb_extra_info["E10"].value = num_switches
wb_extra_info["E11"].value = num_switches_lev_1
wb_extra_info["E12"].value = num_switches_lev_2
wb_extra_info["E13"].value = num_terminals
wb_extra_info["E14"].value = number_of_levels
wb_extra_info["E15"].value = most_populated_level

```

```

#Depends on the number of ranges
num_ranges = len(ranges_list)
j = 0
for col in range(5, 5 + num_ranges):
    name = ranges_list[j]
    wb_extra_info.cell(row=16, column = col).value = name
    wb_extra_info.cell(row=17, column = col).value = ranges_dict[name]
    j += 1

#F. Interoperability
wb_extra_info["E21"].value = manufacturer_concentrator
#wb_extra_info["E22"].value =
#Depends on the number of manufacturers
num_manufacturers = len(manufacturer_list)
i = 0
for col in range(5, 5 + num_manufacturers):
    manufacturer = manufacturer_list[i]
    wb_extra_info.cell(row=23, column = col).value = manufacturer
    wb_extra_info.cell(row=24, column = col).value =
get_manufacturer_summary_value(df_summary, manufacturer, "Topology", "Connected
Nodes")
    wb_extra_info.cell(row=25, column = col).value =
manufacturer_percentage_dict[manufacturer] / 100
    wb_extra_info.cell(row=26, column = col).value =
get_manufacturer_summary_value(df_summary, manufacturer, "Topology", "Average
Availability")
    wb_extra_info.cell(row=27, column = col).value =
get_manufacturer_summary_value(df_summary, manufacturer, "Short Cycles", "Average
Availability")
    wb_extra_info.cell(row=28, column = col).value =
get_manufacturer_summary_value(df_summary, manufacturer, "Long Cycles", "Average
Availability")
    wb_extra_info.cell(row=29, column = col).value =
manufacturer_switches_dict[manufacturer]
    wb_extra_info.cell(row=30, column = col).value =
manufacturer_terminals_dict[manufacturer]
    i += 1

combination12 = ""
combination23 = ""
combination34 = ""
combination45 = ""
combination56 = ""
combination67 = ""
combination78 = ""
combination89 = ""
combination910 = ""

for combination in top_relations_dict["1-2"]:
    if combination12 == "":
        combination12 += combination
    else:
        combination12 = combination12 + " , " + combination
for combination in top_relations_dict["2-3"]:
    if combination23 == "":
        combination23 += combination
    else:
        combination23 = combination23 + " , " + combination
for combination in top_relations_dict["3-4"]:
    if combination34 == "":

```

```

        combination34 += combination
    else:
        combination34 = combination34 + " , " + combination
for combination in top_relations_dict["4-5"]:
    if combination45 == "":
        combination45 += combination
    else:
        combination45 = combination45 + " , " + combination
for combination in top_relations_dict["5-6"]:
    if combination56 == "":
        combination56 += combination
    else:
        combination56 = combination56 + " , " + combination
for combination in top_relations_dict["6-7"]:
    if combination67 == "":
        combination67 += combination
    else:
        combination67 = combination67 + " , " + combination
for combination in top_relations_dict["7-8"]:
    if combination78 == "":
        combination78 += combination
    else:
        combination78 = combination78 + " , " + combination
for combination in top_relations_dict["8-9"]:
    if combination89 == "":
        combination89 += combination
    else:
        combination89 = combination89 + " , " + combination
for combination in top_relations_dict["9-10"]:
    if combination910 == "":
        combination910 += combination
    else:
        combination910 = combination910 + " , " + combination

wb_extra_info["E32"].value = combination12
wb_extra_info["F32"].value = combination23
wb_extra_info["G32"].value = combination34
wb_extra_info["H32"].value = combination45
wb_extra_info["I32"].value = combination56
wb_extra_info["J32"].value = combination67
wb_extra_info["K32"].value = combination78
wb_extra_info["L32"].value = combination89
wb_extra_info["M32"].value = combination910

#Fill the "METER CONCENTRATION" sheet (Optional)

if cgp_data == 1:

    #Open the "METER CONCENTRATION", "NODS_STATES" and "SUMMARY" sheets of
the powerful tool Excel File
    wb_meter_concentration = wb_power["METER_CONCENTRATION_PY"]
    wb_nods_states = wb_power["NODS_STATES"]
    wb_summary = wb_power["SUMMARY"]

    #Fill the table (at least the essential fields)
    #Select the dataframe corresponding to this CT
    id_CT = df_summary.iloc[0,0]
    df_CT = df_STG[df_STG["COD_CT"] == id_CT]

    #Extract the CGP ids corresponding to that CT

```

```

cgp_array = df_CT.loc[:, "COD_CGP"].to_numpy(dtype=int)
#List with the CGP names without repetition
cgp_list = []
for cgp in cgp_array:
    if cgp not in cgp_list:
        cgp_list.append(cgp)
#Iteration for every CGP id in the list
i = 2
n_meters_found_array = []
n_meters_not_found_array = []
for cgp in cgp_list:
    df_cgp = df_CT[df_CT["COD_CGP"] == cgp]
    meter_array = df_cgp.loc[:, "COD_METER"].to_numpy() #Get the array
with the meters present in the database
#Check that the STG database meters are present in the PRIME
subnetwork
    address_name = str(df_cgp.iloc[0,6]) + "-" + str(df_cgp.iloc[0,1]) +
    "-" + str(df_cgp.iloc[0,9]) + "-" + str(df_cgp.iloc[0,2]) + "-" +
df_cgp.iloc[0,5]
    meters_found = 0
    meters_not_found = 0
    avg_avail = []
    avg_avail_short = []
    avg_avail_long = []
    for meter in meter_array:
        #Look for the meter on the STG database in the "NODS_STATES"
sheet
        df_meter = df_nodes_states[df_nodes_states["Meter"] == meter]
        if len(df_meter) == 1: #Meter found
            meters_found += 1
            avg_avail.append(df_meter.iloc[0,5])
            avg_avail_short.append(df_meter.iloc[0,18])
            avg_avail_long.append(df_meter.iloc[0,22])
        else: #Meter not found
            meters_not_found += 1
    if meters_found > 0: #Fill only if that CGP belongs to any meter of
the powerful tool Excel file
        #Calculate the average values
        cgp_avail = sum(avg_avail) / len(avg_avail)
        cgp_avail_short = sum(avg_avail_short) / len(avg_avail_short)
        cgp_avail_long = sum(avg_avail_long) / len(avg_avail_long)
        #Fill the Excel file
        wb_meter_concentration.cell(row = i, column = 1).value =
address_name
        wb_meter_concentration.cell(row = i, column = 2).value =
meters_found
        wb_meter_concentration.cell(row = i, column = 3).value =
meters_not_found
        wb_meter_concentration.cell(row = i, column = 6).value =
cgp_avail
        wb_meter_concentration.cell(row=i, column=6).number_format =
FORMAT_PERCENTAGE_00
        wb_meter_concentration.cell(row = i, column = 12).value =
cgp_avail_short
        wb_meter_concentration.cell(row=i, column=12).number_format =
FORMAT_PERCENTAGE_00
        wb_meter_concentration.cell(row = i, column = 13).value =
cgp_avail_long
        wb_meter_concentration.cell(row=i, column=13).number_format =
FORMAT_PERCENTAGE_00

```

```

        wb_meter_concentration.cell(row = i, column = 18).value = cgp
        i += 1
        n_meters_found_array.append(meters_found)
        n_meters_not_found_array.append(meters_not_found)
#Look for the meters of the PRIME subnetwork in STG.
#Get the meter list from the "NODS_STATES" sheet
all_meters_array = df_nodes_states.loc[:, "Meter"].to_numpy()
all_meters_list = []
for meter in all_meters_array:
    all_meters_list.append(meter)
#Look for the address in STG
i = 2
network_meters_in_database_same_ct = 0
network_meters_in_database_other_ct = 0
network_meters_not_in_database = 0
meters_this_ct = []
for meter in all_meters_list:
    if meter is not np.nan:
        df_this_meter = df_STG[df_STG["COD_METER"] == meter]
        if len(df_this_meter) == 1:
            if df_this_meter.iloc[0,0] == id_CT:
                network_meters_in_database_same_ct += 1
                meters_this_ct.append(meter)
            else:
                network_meters_in_database_other_ct += 1
        wb_nods_states.cell(row=i, column=26).value =
df_this_meter.iloc[0,5] #Fill the address in the "NODS_STATES" sheet
        else:
            network_meters_not_in_database += 1
            meters_this_ct.append(meter)
        i += 1

#Save results of the availabilities of only the meters that correspond to
this ct
wb_extra_info.cell(row=4, column=11).value =
my_get_availabilities_meters_this_ct(df_nodes_states,
meters_this_ct, "Topology", "Average")
wb_extra_info.cell(row=4, column=12).value =
my_get_availabilities_meters_this_ct(df_nodes_states,
meters_this_ct, "Topology", "Standard Deviation")
wb_extra_info.cell(row=5, column=11).value =
my_get_availabilities_meters_this_ct(df_nodes_states, meters_this_ct, "Short
Cycles", "Average")
wb_extra_info.cell(row=5, column=12).value =
my_get_availabilities_meters_this_ct(df_nodes_states, meters_this_ct, "Short
Cycles", "Standard Deviation")
wb_extra_info.cell(row=6, column=11).value =
my_get_availabilities_meters_this_ct(df_nodes_states, meters_this_ct, "Long
Cycles", "Average")
wb_extra_info.cell(row=6, column=12).value =
my_get_availabilities_meters_this_ct(df_nodes_states, meters_this_ct, "Long
Cycles", "Standard Deviation")

#Save the results in the "SUMMARY" sheet
wb_summary.cell(row=6, column=4).value = sum(n_meters_found_array)
wb_summary.cell(row=7, column=4).value = sum(n_meters_not_found_array)
wb_summary.cell(row=10, column=4).value =
network_meters_in_database_same_ct
wb_summary.cell(row=11, column=4).value =
network_meters_in_database_other_ct

```

```
wb_summary.cell(row=12, column=4).value = network_meters_not_in_database

#Save the changes on the Excel file
wb_power.save(power_filename)

print("Processing finished.")
```

SCRIPT 4: PLC PERFORMANCE RESULTS CONCENTRATOR SCRIPT

```
import numpy as np
import pandas as pd
import os
import openpyxl as op
from openpyxl.styles.numbers import FORMAT_PERCENTAGE_00

#####
##### Functions #####
#####

def my_get_summary_connected_nodes(df_ev_nods_topo):
    """
    Calculates the statistics of the connected nodes during the monitoring.

    Parameters:
        df_ev_nods_topo (Pandas Dataframe): Dataframe of the "EV_NODS_TOPO" sheet
of the
        powerful tool Excel file.

    Returns:
        connected_nodes_dict (dict): Dictionary with the statistics as keys.
    """

    #Create the dictionary
    connected_nodes_dict = {"Average":0, "Standard_deviation":0, "Max":0,
"Min":0}

    #Fill the dictionary
    connected_nodes_dict["Average"] = df_ev_nods_topo.iloc[0,6]
    node_series = df_ev_nods_topo.iloc[:,1].to_numpy(dtype=int)
    connected_nodes_dict["Standard_deviation"] = np.nanstd(node_series)
    connected_nodes_dict["Max"] = np.nanmax(node_series)
    connected_nodes_dict["Min"] = np.nanmin(node_series)

    return connected_nodes_dict

def my_get_summary_switches(df_ev_swts_topo):
    """
    Calculates the statistics of the connected switches during the monitoring.

    Parameters:
        df_ev_swts_topo (Pandas Dataframe): Dataframe of the "EV_SWTS_TOPO" sheet
of the
        powerful tool Excel file.

    Returns:
        connected_switches_dict (dict): Dictionary with the statistics as keys.
```

```

'''

#Create the dictionary
connected_switches_dict = {"Average":0, "Standard_deviation":0, "Max":0,
"Min":0}

#Fill the dictionary
connected_switches_dict["Average"] = df_ev_swts_topo.iloc[0,6]
switch_series = df_ev_swts_topo.iloc[:,1].to_numpy(dtype=int)
connected_switches_dict["Standard_deviation"] = np.nanstd(switch_series)
connected_switches_dict["Max"] = np.nanmax(switch_series)
connected_switches_dict["Min"] = np.nanmin(switch_series)

return connected_switches_dict

def my_get_summary_reading_short(df_ev_short):
'''
Calculates the statistics of the short cycles during the monitoring.

Parameters:
    df_ev_short (Pandas Dataframe): Dataframe of the "EV_SHORT" sheet of the
    powerful tool Excel file.

Returns:
    reading_short_dict (dict): Dictionary with the statistics as keys.
'''

#Create the dictionary
reading_short_dict = {"Average":0, "Standard_deviation":0, "Max":0, "Min":0}
short_series = df_ev_short.loc[:, "Duración ciclo (seg)"].to_numpy(dtype=int)
reading_short_dict["Average"] = np.nanmean(short_series)
reading_short_dict["Standard_deviation"] = np.nanstd(short_series)
reading_short_dict["Max"] = np.nanmax(short_series)
reading_short_dict["Min"] = np.nanmin(short_series)

return reading_short_dict

def my_get_summary_reading_long(df_ev_long):
'''
Calculates the statistics of the long cycles during the monitoring.

Parameters:
    df_ev_long (Pandas Dataframe): Dataframe of the "EV_LONG" sheet of the
    powerful
    tool Excel file.

Returns:
    reading_long_dict (dict): Dictionary with the statistics as keys.
'''

#Create the dictionary
reading_long_dict = {"Average":0, "Standard_deviation":0, "Max":0, "Min":0}
long_series = df_ev_long.loc[:, "Duración ciclo (seg)"].to_numpy(dtype=int)
reading_long_dict["Average"] = np.nanmean(long_series)
reading_long_dict["Standard_deviation"] = np.nanstd(long_series)
reading_long_dict["Max"] = np.nanmax(long_series)
reading_long_dict["Min"] = np.nanmin(long_series)

return reading_long_dict

```

```
def my_get_summary_coms_availability(df_ev_nods_states):
    """
    Calculates the statistics of the topology availability during the monitoring.

    Parameters:
        df_ev_nods_states (Pandas Dataframe): Dataframe of the "NODS_STATES"
sheet of
        the powerful tool Excel file.

    Returns:
        com_avail_dict (dict): Dictionary with the statistics as keys.
    """

    #Create the dictionary
    com_avail_dict = {"Average":0, "Standard_deviation":0, "Max":0, "Min":0}
    com_series = df_ev_nods_states.loc[:, "Available"].to_numpy(dtype=float)
    com_avail_dict["Average"] = np.nanmean(com_series)
    com_avail_dict["Standard_deviation"] = np.nanstd(com_series)
    com_avail_dict["Max"] = np.nanmax(com_series)
    com_avail_dict["Min"] = np.nanmin(com_series)

    return com_avail_dict

#####
##### Main #####
#####

#Open the CT summary Excel file
summary_filename = "monitoring_summary.xlsx"
wb_power = op.load_workbook(summary_filename)
wb_sheet = wb_power["Data"]

#Iterate all the processed Powerful tool Excel files
directory = os.path.join(os.getcwd(), "Ficheros_concentrar")

for file in os.listdir(directory):
    #Read the processed Powerful tool Excel file
    power_file = os.path.join(directory, file)
    print(power_file)
    df_summary = pd.read_excel(power_file, sheet_name="SUMMARY")
    df_extra_info = pd.read_excel(power_file, sheet_name="EXTRA_INFORMATION")
    df_ev_nods_topo = pd.read_excel(power_file, sheet_name="EV_NODS_TOPO",
header=None)
    df_ev_swts_topo = pd.read_excel(power_file, sheet_name="EV_SWTS_TOPO",
header=None)
    df_ev_short = pd.read_excel(power_file, sheet_name="EV_SHORT")
    df_ev_long = pd.read_excel(power_file, sheet_name="EV_LONG")
    df_ev_nods_states = pd.read_excel(power_file, sheet_name="NODS_STATES")

    #Fill the CT summary Excel file with the info from the Powerful tool Excel
file
    ct_id = df_summary.iloc[0,0]
    connected_nodes_dict = my_get_summary_connected_nodes(df_ev_nods_topo)
    connected_switches_dict = my_get_summary_switches(df_ev_swts_topo)
    reading_short_dict = my_get_summary_reading_short(df_ev_short)
    reading_long_dict = my_get_summary_reading_long(df_ev_long)
    com_avail_dict = my_get_summary_coms_availability(df_ev_nods_states)

    #Iterate the rows of the summary table until the CT id matches
    for i in range(6, wb_sheet.max_row + 1):
```

```

if wb_sheet.cell(row=i, column=1).value == ct_id:

    wb_sheet.cell(row=i, column=11).value = df_summary.iloc[4,3]
    wb_sheet.cell(row=i, column=12).value = df_summary.iloc[5,3]
    wb_sheet.cell(row=i, column=13).value = df_summary.iloc[8,3]
    wb_sheet.cell(row=i, column=14).value = df_summary.iloc[9,3]
    wb_sheet.cell(row=i, column=15).value = df_summary.iloc[10,3]

    wb_sheet.cell(row=i, column=16).value = df_extra_info.iloc[2,4]
#A.1.2

    #A.1.3
    wb_sheet.cell(row=i, column=17).value =
connected_nodes_dict["Average"]
    wb_sheet.cell(row=i, column=18).value =
connected_nodes_dict["Standard deviation"]
    wb_sheet.cell(row=i, column=19).value = connected_nodes_dict["Max"]
    wb_sheet.cell(row=i, column=20).value = connected_nodes_dict["Min"]

    wb_sheet.cell(row=i, column=21).value =
connected_switches_dict["Average"] #A.1.4
    wb_sheet.cell(row=i, column=22).value =
connected_switches_dict["Standard deviation"]
    wb_sheet.cell(row=i, column=23).value =
connected_switches_dict["Max"]
    wb_sheet.cell(row=i, column=24).value =
connected_switches_dict["Min"]

    wb_sheet.cell(row=i, column=25).value = df_extra_info.iloc[3,5]
#A.2.2

    wb_sheet.cell(row=i, column=26).value = df_extra_info.iloc[3,6]
    wb_sheet.cell(row=i, column=27).value = df_extra_info.iloc[3,7]
    wb_sheet.cell(row=i, column=28).value = df_extra_info.iloc[3,8]
    wb_sheet.cell(row=i, column=29).value = df_extra_info.iloc[3,10]
    wb_sheet.cell(row=i, column=30).value = df_extra_info.iloc[3,11]
    wb_sheet.cell(row=i, column=25).number_format = FORMAT_PERCENTAGE_00
    wb_sheet.cell(row=i, column=26).number_format = FORMAT_PERCENTAGE_00
    wb_sheet.cell(row=i, column=27).number_format = FORMAT_PERCENTAGE_00
    wb_sheet.cell(row=i, column=28).number_format = FORMAT_PERCENTAGE_00
    wb_sheet.cell(row=i, column=29).number_format = FORMAT_PERCENTAGE_00
    wb_sheet.cell(row=i, column=30).number_format = FORMAT_PERCENTAGE_00

    wb_sheet.cell(row=i, column=31).value = reading_short_dict["Average"]
#A.2.4

    wb_sheet.cell(row=i, column=32).value =
reading_short_dict["Standard deviation"]
    wb_sheet.cell(row=i, column=33).value = reading_short_dict["Max"]
    wb_sheet.cell(row=i, column=34).value = reading_short_dict["Min"]

    wb_sheet.cell(row=i, column=35).value = df_extra_info.iloc[4,5]
#A.3.2

    wb_sheet.cell(row=i, column=36).value = df_extra_info.iloc[4,6]
    wb_sheet.cell(row=i, column=37).value = df_extra_info.iloc[4,7]
    wb_sheet.cell(row=i, column=38).value = df_extra_info.iloc[4,8]
    wb_sheet.cell(row=i, column=39).value = df_extra_info.iloc[4,10]
    wb_sheet.cell(row=i, column=40).value = df_extra_info.iloc[4,11]
    wb_sheet.cell(row=i, column=35).number_format = FORMAT_PERCENTAGE_00
    wb_sheet.cell(row=i, column=36).number_format = FORMAT_PERCENTAGE_00
    wb_sheet.cell(row=i, column=37).number_format = FORMAT_PERCENTAGE_00
    wb_sheet.cell(row=i, column=38).number_format = FORMAT_PERCENTAGE_00

```

```

wb_sheet.cell(row=i, column=39).number_format = FORMAT_PERCENTAGE_00
wb_sheet.cell(row=i, column=40).number_format = FORMAT_PERCENTAGE_00

wb_sheet.cell(row=i, column=41).value = reading_long_dict["Average"]
#A.3.4
wb_sheet.cell(row=i, column=42).value =
reading_long_dict["Standard_deviation"]
wb_sheet.cell(row=i, column=43).value = reading_long_dict["Max"]
wb_sheet.cell(row=i, column=44).value = reading_long_dict["Min"]

wb_sheet.cell(row=i, column=45).value = com_avail_dict["Average"]
#B.1.2
wb_sheet.cell(row=i, column=46).value =
com_avail_dict["Standard_deviation"]
wb_sheet.cell(row=i, column=47).value = com_avail_dict["Max"]
wb_sheet.cell(row=i, column=48).value = com_avail_dict["Min"]
wb_sheet.cell(row=i, column=49).value = df_extra_info.iloc[2,10]
wb_sheet.cell(row=i, column=50).value = df_extra_info.iloc[2,11]
wb_sheet.cell(row=i, column=45).number_format = FORMAT_PERCENTAGE_00
wb_sheet.cell(row=i, column=46).number_format = FORMAT_PERCENTAGE_00
wb_sheet.cell(row=i, column=47).number_format = FORMAT_PERCENTAGE_00
wb_sheet.cell(row=i, column=48).number_format = FORMAT_PERCENTAGE_00
wb_sheet.cell(row=i, column=49).number_format = FORMAT_PERCENTAGE_00
wb_sheet.cell(row=i, column=50).number_format = FORMAT_PERCENTAGE_00

wb_sheet.cell(row=i, column=51).value = df_extra_info.iloc[8,4]
#C.1.2
wb_sheet.cell(row=i, column=52).value = df_extra_info.iloc[9,4]
#C.1.3
wb_sheet.cell(row=i, column=53).value = df_extra_info.iloc[10,4]
#C.1.4
wb_sheet.cell(row=i, column=54).value = df_extra_info.iloc[11,4]
#C.1.5
wb_sheet.cell(row=i, column=55).value = df_extra_info.iloc[12,4]
#C.1.6
wb_sheet.cell(row=i, column=56).value = df_extra_info.iloc[13,4]
#C.1.7

wb_sheet.cell(row=i, column=57).value = df_extra_info.iloc[15,13]
#C.1.8
wb_sheet.cell(row=i, column=58).value = df_extra_info.iloc[15,12]
wb_sheet.cell(row=i, column=59).value = df_extra_info.iloc[15,11]
wb_sheet.cell(row=i, column=60).value = df_extra_info.iloc[15,10]
wb_sheet.cell(row=i, column=61).value = df_extra_info.iloc[15,9]
wb_sheet.cell(row=i, column=62).value = df_extra_info.iloc[15,8]
wb_sheet.cell(row=i, column=63).value = df_extra_info.iloc[15,7]
wb_sheet.cell(row=i, column=64).value = df_extra_info.iloc[15,6]
wb_sheet.cell(row=i, column=65).value = df_extra_info.iloc[15,5]
wb_sheet.cell(row=i, column=66).value = df_extra_info.iloc[15,4]

wb_sheet.cell(row=i, column=67).value = df_extra_info.iloc[19,4]
#F.1.1
wb_sheet.cell(row=i, column=68).value = df_extra_info.iloc[20,4]
#F.1.2

#Depends on the manufacturers present
#F.2.2, #F.2.3, #F.3.1, #F.3.2, #F.3.3, #F.4.1, #F.4.2
manufacturer_position_dict = {"ZIV":1, "Landis":2, "SAGEM":3,
"SOGECAM":4, "ORBIS":5,

```

```

        "CIRCUTOR":6, "ELSTER":7, "GEE":8, "ITRON":9)
    manufacturer_list = ["ZIV", "Landis", "SAGEM", "SOGECAM", "ORBIS",
"CIRCUTOR", "ELSTER"
, "GEE", "ITRON"]
    for manufacturer in manufacturer_list:
        #Look if the manufacturer appears in the "EXTRA INFORMATION"
sheet
        df_manufacturer = df_extra_info.iloc[21:29,:]
        initial_pos = 68 + manufacturer_position_dict[manufacturer]
        a = 4
        for k in range(4,14): #Look for the column corresponding to
the manufacturer
            if df_manufacturer.iloc[0,k] == manufacturer:
                break
            else:
                a += 1
            if a < 14: #Manufacturer found
                wb_sheet.cell(row=i, column=initial_pos).value =
df_manufacturer.iloc[1,a] #F.2.2
                wb_sheet.cell(row=i, column=initial_pos + 9*1).value =
df_manufacturer.iloc[2,a] #F.2.3
                wb_sheet.cell(row=i, column=initial_pos + 9*1).number_format
= FORMAT_PERCENTAGE_00
                wb_sheet.cell(row=i, column=initial_pos + 9*2).value =
df_manufacturer.iloc[3,a] #F.3.1
                wb_sheet.cell(row=i, column=initial_pos + 9*2).number_format
= FORMAT_PERCENTAGE_00
                wb_sheet.cell(row=i, column=initial_pos + 9*3).value =
df_manufacturer.iloc[4,a] #F.3.2
                wb_sheet.cell(row=i, column=initial_pos + 9*3).number_format
= FORMAT_PERCENTAGE_00
                wb_sheet.cell(row=i, column=initial_pos + 9*4).value =
df_manufacturer.iloc[5,a] #F.3.3
                wb_sheet.cell(row=i, column=initial_pos + 9*4).number_format
= FORMAT_PERCENTAGE_00
                wb_sheet.cell(row=i, column=initial_pos + 9*5).value =
df_manufacturer.iloc[6,a] #F.4.1
                wb_sheet.cell(row=i, column=initial_pos + 9*6).value =
df_manufacturer.iloc[7,a] #F.4.2
            else: #Manufacturer not found
                wb_sheet.cell(row=i, column=initial_pos).value = 0
#F.2.2
                wb_sheet.cell(row=i, column=initial_pos + 9*1).value = 0
#F.2.3
                wb_sheet.cell(row=i, column=initial_pos + 9*1).number_format
= FORMAT_PERCENTAGE_00
                wb_sheet.cell(row=i, column=initial_pos + 9*2).value = 0
#F.3.1
                wb_sheet.cell(row=i, column=initial_pos + 9*2).number_format
= FORMAT_PERCENTAGE_00
                wb_sheet.cell(row=i, column=initial_pos + 9*3).value = 0
#F.3.2
                wb_sheet.cell(row=i, column=initial_pos + 9*3).number_format
= FORMAT_PERCENTAGE_00
                wb_sheet.cell(row=i, column=initial_pos + 9*4).value = 0
#F.3.3
                wb_sheet.cell(row=i, column=initial_pos + 9*4).number_format
= FORMAT_PERCENTAGE_00
                wb_sheet.cell(row=i, column=initial_pos + 9*5).value = 0
#F.4.1

```

```

wb_sheet.cell(row=i, column=initial_pos + 9*6).value = 0
#F.4.2

wb_sheet.cell(row=i, column=132).value = df_extra_info.iloc[30,4]
#F.4.3
wb_sheet.cell(row=i, column=133).value = df_extra_info.iloc[30,5]
wb_sheet.cell(row=i, column=134).value = df_extra_info.iloc[30,6]
wb_sheet.cell(row=i, column=135).value = df_extra_info.iloc[30,7]
wb_sheet.cell(row=i, column=136).value = df_extra_info.iloc[30,8]
wb_sheet.cell(row=i, column=137).value = df_extra_info.iloc[30,9]
wb_sheet.cell(row=i, column=138).value = df_extra_info.iloc[30,10]
wb_sheet.cell(row=i, column=139).value = df_extra_info.iloc[30,11]
wb_sheet.cell(row=i, column=140).value = df_extra_info.iloc[30,12]

wb_sheet.cell(row=i, column=141).value = df_extra_info.iloc[34,4]
#G.1
wb_sheet.cell(row=i, column=142).value = df_extra_info.iloc[35,4]
#G.2
wb_sheet.cell(row=i, column=143).value = df_extra_info.iloc[36,4]
#G.3
wb_sheet.cell(row=i, column=144).value = df_extra_info.iloc[37,4]
#G.4
wb_sheet.cell(row=i, column=145).value = df_extra_info.iloc[38,4]
#G.5
wb_sheet.cell(row=i, column=146).value = df_extra_info.iloc[39,4]
#G.6

break

wb_power.save(summary_filename)
print("Processing finished.")

```

SCRIPT 5: MATLAB CLUSTERING SCRIPT

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Input parameters %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Choose from where the data wants to be obtained for the clustering.
% 1 --> Topology histogram (C.1.9)
% 2 --> Cycle and comms availabilities (avg and SD)
% 3 --> Option 2 + Connected nodes (avg and SD)
% 4 --> Option 1 + Option 2
% 5 --> All

option = 4;

if option == 1
    data = readtable('cluster_data.xlsx','Sheet','Option 1','Range','A2:J146');
elseif option == 2
    data = readtable('cluster_data.xlsx','Sheet','Option 2','Range','A2:F146');
elseif option == 3
    data = readtable('cluster_data.xlsx','Sheet','Option 3','Range','A2:H146');
elseif option == 4
    data = readtable('cluster_data.xlsx','Sheet','Option 4','Range','A2:J146');
elseif option == 5
    data = readtable('cluster_data.xlsx','Sheet','Option 5','Range','A2:P146');
elseif option == 6

```

```

data = readtable('cluster_data.xlsx','Sheet','Option 6','Range','A2:R146');
end

data2 = table2array(data); %Convert the data from table to array

%Scale the registered nodes and switches variables (only for options 3,4
%and 6)
%Min-max normalization
if option == 3 || option == 4 || option == 6
    if option == 3 || option == 6
        b = 2;
    else
        b = 4;
    end
    for i = 1:b
        vector_not_scaled = data2(:,i);
        maximum = max(vector_not_scaled);
        minimum = min(vector_not_scaled);
        vector_scaled = (vector_not_scaled - minimum) / (maximum - minimum) ;
        data2(:,i) = vector_scaled;
    end
end

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Algorithm %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%K-means clustering
opts = statset('Display','final');
k = 6; %Number of clusters
[idx,C,sumd,D] =
kmeans(data2,k,'Distance','sqeuclidean','Replicates',30,'Options',opts);

% Calculate the number of elements belonging to each cluster
elements_per_group = zeros(k,1);
for j = 1:k
    elements_per_group(j,1) = sum(idx == j);
end

%Dimensionality reduction with PCA for cluster visualization
[coeff,score,latent,tsquared,explained] = pca(data2,'Centered',false);
% Transform the centroids to Principal Components
C_pca = C * coeff;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Plots %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% 3D plot
figure;
plot3(score(idx==1,1),score(idx==1,2),score(idx==1,3),'r.','MarkerSize',12)
hold on
plot3(score(idx==2,1),score(idx==2,2),score(idx==2,3),'g.','MarkerSize',12)
plot3(score(idx==3,1),score(idx==3,2),score(idx==3,3),'m.','MarkerSize',12)
plot3(score(idx==4,1),score(idx==4,2),score(idx==4,3),'b.','MarkerSize',12)
plot3(score(idx==5,1),score(idx==5,2),score(idx==5,3),'c.','MarkerSize',12)
plot3(score(idx==6,1),score(idx==6,2),score(idx==6,3),'y.','MarkerSize',12)
plot3(score(idx==7,1),score(idx==7,2),score(idx==7,3),'k.','MarkerSize',12)
plot3(C_pca(:,1),C_pca(:,2),C_pca(:,3),'kx','MarkerSize',15,'LineWidth',3)

```

```

if k == 4
    legend('Cluster 1','Cluster 2','Cluster 3','Cluster
4','Centroids','Location','NW')
elseif k == 5
    legend('Cluster 1','Cluster 2','Cluster 3','Cluster 4','Cluster
5','Centroids','Location','NW')
elseif k == 6
    legend('Cluster 1','Cluster 2','Cluster 3','Cluster 4','Cluster 5','Cluster
6','Centroids','Location','NW')
elseif k == 7
    legend('Cluster 1','Cluster 2','Cluster 3','Cluster 4','Cluster 5','Cluster
6','Cluster 7','Centroids','Location','NW')
end
title('Cluster Assignments and Centroids')
grid on
hold off

%2D plot
figure;
plot(score(idx==1,1),score(idx==1,2),'r.','MarkerSize',12)
hold on
plot(score(idx==2,1),score(idx==2,2),'g.','MarkerSize',12)
plot(score(idx==3,1),score(idx==3,2),'m.','MarkerSize',12)
plot(score(idx==4,1),score(idx==4,2),'b.','MarkerSize',12)
plot(score(idx==5,1),score(idx==5,2),'c.','MarkerSize',12)
plot(score(idx==6,1),score(idx==6,2),'y.','MarkerSize',12)
plot(score(idx==7,1),score(idx==7,2),'k.','MarkerSize',12)
plot(C_pca(:,1),C_pca(:,2),'kx','MarkerSize',15,'LineWidth',3)
legend('Cluster 1','Cluster 2','Cluster 3','Cluster
4','Centroids','Location','NW')
if k == 4
    legend('Cluster 1','Cluster 2','Cluster 3','Cluster
4','Centroids','Location','NW')
elseif k == 5
    legend('Cluster 1','Cluster 2','Cluster 3','Cluster 4','Cluster
5','Centroids','Location','NW')
elseif k == 6
    legend('Cluster 1','Cluster 2','Cluster 3','Cluster 4','Cluster 5','Cluster
6','Centroids','Location','NW')
elseif k == 7
    legend('Cluster 1','Cluster 2','Cluster 3','Cluster 4','Cluster 5','Cluster
6','Cluster 7','Centroids','Location','NW')
end
title('Cluster Assignments and Centroids')
grid on
hold off

%Cluster number (k) comparison
k_v = 1:20;
suma = zeros(length(k_v),1);
for k_iter = 1:length(k_v)
    k = k_v(k_iter);
    [idx2,C2,sumd2,D2] =
kmeans(data2,k,'Distance','sqeuclidean','Replicates',15,'Options',opts);
    suma(k_iter) = sum(sumd2);
end
figure
bar(k_v,suma)
%plot(k_v,suma,,'.-')
title('Clustering Option ' + string(option))

```

```
xlabel('Number of clusters (K)')  
ylabel('Sum of centroid distances')
```