

INFORME CUARTO  
SPRINT CHEC



## **1. Depuración de bugs e incorporación de mejoras en el algoritmo de análisis nodal y de potencias, construido en sprint 3.**

Durante el desarrollo de este sprint, y el comienzo de la ejecución de pruebas del algoritmo de reglas impuestas desarrollado en el sprint anterior; se pudo evidenciar que dicho algoritmo presentaba algunos bugs; por tanto, se presentó la necesidad de dar corrección a estos. Sumado al hecho que durante la revisión detalla de los resultados del algoritmo se pudo evidenciar ciertas mejoras que se podían implementar para aumentar la eficiencia de este, y reducir a su vez en ciertos aspectos el costo computacional que podría presentarse, además de dar más fiabilidad al mismo.

### **Cambio de optimización:**

- Imputamos valores de Corrientes y voltajes según las potencias: Se analizan los casos en los cuales la potencia activa y reactiva son cero, en estos casos podemos analizar dos posibles escenarios:
  1. Si las corrientes en las tres fases no son nulas, y además son diferentes de cero, entonces en este caso, se puede tener la certeza, de que los voltajes en las tres fases pueden ser reemplazados por cero, independientemente si el valor original está perdido o no.
  2. Si los voltajes en las tres fases no son nulos, y además son diferentes de cero, entonces en este caso, se puede tener la certeza, de que las corrientes en las tres fases pueden ser reemplazadas por cero, independientemente si el valor original está perdido o no.
- Análisis nodal en las barras de más de tres circuitos: Teniendo en cuenta que, para los registros actuales, las potencias se están registrando con su signo correctamente, entonces, teniendo el signo de la potencia activa para cada uno de los circuitos de la barra, se tiene la certeza, de que está entrando y que está saliendo; permitiendo esto hacer un análisis nodal, más confiable. Además, este cambio elimina el hecho de tener que hacer todas las combinaciones posibles entre corrientes, para encontrar un análisis nodal adecuado; disminuyendo esto sustancialmente el costo computacional demandado por el algoritmo. Además, en el caso en el cual, el análisis nodal utilizando los signos de estas potencias, no tenga sentido, se procede a generar una alerta indicando que pueden estar sucediendo alguno de los dos siguientes escenarios:
  1. Los signos de las potencias no son correctos.

2. Hay un circuito de la barra que no se está registrando en base de datos.
- **análisis nodal sin fases completas:** Se realizó un ajuste al algoritmo, que permite realizar una imputación de datos de corrientes en caso de que no haya ninguna fase completa, bajo la condición de que solo falte la información de uno de los circuitos, y se tengan todos los signos de las potencias según los últimos registros, ya que basándose en ese flujo es posible tener plena certeza de los valores faltantes en las corrientes del circuito sin información, en caso de que hayan más circuitos con datos perdidos, y se tengan los signos de las potencias, esto ya no es posible porque a pesar de que se tiene certeza en el flujo, no se tendría certeza en la magnitud y sería mejor basarse en un modelo de imputación que en un algoritmo de reglas impuestas. Cabe aclarar que la necesidad que este el signo de la potencia es para tener un flujo claro en la barra, pero para los casos de 3 circuitos y N circuitos, dentro del algoritmo quedó definido un caso especial, donde imputamos corrientes aun sin tener claro el signo de la potencia del circuito faltante, ubicándolo acorde a como este se adapte para completar el análisis nodal, es decir si el flujo que entra a la barra es menor al que sale, el circuito con datos perdidos se acopla a la información faltante y viceversa.
  - **Corrección del algoritmo caso específico de fases completas:** Observamos un problema de fiabilidad dentro del algoritmo de reglas impuestas, donde al analizar fases completas para comprobar el flujo dado por los signos de las potencias de los últimos registros, si dicha fase completa para comprobar es cero en todos sus valores, no debería ser un criterio fiable para definir si el flujo es correcto para imputar, puesto que el circuito estaría apagado, y el error en dicho caso sería engañoso demostrando un valor ideal de error como 0%, cuando realmente no necesariamente.
  - **Paralelización del algoritmo:** Teniendo en cuenta, que el procesamiento de cada instante de tiempo, de una determinada barra, es independiente de los demás instantes de tiempo, se decidió, paralelizar el algoritmo de tal forma, que este pueda realizar el respectivo análisis nodal y de potencias, para múltiples instantes de tiempo, a la vez, de una determinada barra, dependiendo del número de núcleos de la máquina en la cual se ejecute el algoritmo. Permitiendo esto disminuir considerablemente, el tiempo de ejecución del algoritmo.

Realizando pruebas para una semana de registros con los circuitos del ABB, se obtuvieron los siguientes resultados de tiempos:



Barra	Código estándar (M)	Código paralelizado (M)
TEB40	4.324	1.232
GTO23	7.313	2.532
HER40	12.534	4.542
ENE46	6.765	2.432
SCH30	4.532	1.654
PSO40	6.354	2.123
MAZ23	7.132	2.312
REG30	8.341	3.321
VCT23	6.123	2.863
MAN23	10.233	3.234
QHI23	7.123	2.141

Durante el ejercicio se pudo comprobar una mejora considerable en los tiempos del algoritmo, aproximadamente triplicando su velocidad de cómputo, aclarando que esto puede variar según la cantidad de núcleos del servidor en el cual se despliegue el algoritmo, así como la forma en la cual estos núcleos comparten los recursos globales de la máquina.

Enlace del algoritmo en diagrama esquemático actualizado:

<https://app.diagrams.net/#G1-opOSHv3ZQ-Zs-PVwjpQb7h5lrMq5Z05>

## 2. Resultados temporales para cada una de las barras:

Como la idea es que el algoritmo impute los datos de manera semanal, se realizó un estudio de los tiempos estimados que se tardaría la imputación solo con la estructura de reglas impuestas construida, teniendo en cuenta un caso no tan óptimo utilizando ocho núcleos para el procesamiento, definiendo unos tiempos totales tanto para ABB como SURVALENT que se presentan a continuación:

SISTEMA SCADA	TIEMPOS
ABB	6 horas
SURVALENT	5,2 horas
Total	11,2 horas

Lo anterior teniendo en cuenta, que este procesamiento se realizó incluyendo circuitos que están repetidos en ambos sistemas, por lo tanto estos tiempos se podrían disminuir.

### 3. Resultados de algoritmo de análisis nodal y de potencias sobre circuitos, de SCADA ABB y SCADA SURVALENT.

Una vez ejecutado el algoritmo para todos los circuitos de ambos sistemas durante un periodo de una semana, se pudo evidenciar algunos casos, y sacar las siguientes conclusiones:

El algoritmo hasta el momento logra un gran rendimiento para la mayoría de los circuitos, definiendo un rendimiento por medio de las siguientes tablas:

#### ABB

Variable	Datos perdidos antes	Datos perdidos después	%Imputado
IA	24845	14744	40.1%
IB	31790	21687	31.8%
IC	31788	21685	31.8%
VA	144760	7233	95%
VB	156685	10586	93.1%
VC	158197	12599	92.1%
P	38520	20531	46.7%
Q	37175	19186	48.4%

#### SURVALENT

Variable	Datos perdidos antes	Datos perdidos después	%Imputado
IA	26120	25277	2.3%
IB	24438	23607	3.4%
IC	24655	23822	3.3%
VA	23979	11854	50.6%
VB	24849	14293	42.5%
VC	24518	12909	47.3%
P	26368	23749	10%
Q	25202	20788	17.5%

Como se puede ver, el rendimiento del algoritmo de reglas impuestas en SCADA SURVALENT es considerablemente menor al rendimiento del algoritmo en SCADA ABB; esto debido a que la mayoría de los circuitos presentes en SCADA SURVALENT son circuitos de red y por ende no se puede hacer un análisis nodal.

Casos donde no es posible imputar:

- Barras con un solo circuito: Para estos casos, teniendo en cuenta que no es posible realizar un análisis nodal y únicamente se observa

la posibilidad de generar un análisis de potencias, en el caso donde tengamos corrientes y voltajes perdidos, no hay manera de recuperar los valores para dichas variables.

	CIRCUITO	TIEMPO_AJUSTADO	IA	IB	IC	VA	VB	VC	P	Q	E <sub>ae</sub>	E <sub>re</sub>	E <sub>ai</sub>	E <sub>ri</sub>	FECHA_CARGA
1	TEB30B20	2020-01-03 13:15:00.000	NULL	NULL	NULL	1.0000000	1.0000000	1.0000000	NULL	NULL	NULL	NULL	NULL	NULL	2020-05-12 18:03:16.140
2	TEB30B20	2020-01-03 12:45:00.000	NULL	NULL	NULL	1.0000000	1.0000000	1.0000000	NULL	NULL	NULL	NULL	NULL	NULL	2020-05-12 18:03:16.140
3	TEB30B20	2020-01-03 12:30:00.000	NULL	NULL	NULL	1.0000000	1.0000000	1.0000000	NULL	NULL	NULL	NULL	NULL	NULL	2020-05-12 18:03:16.140
4	TEB30B20	2020-01-03 12:00:00.000	NULL	NULL	NULL	1.0000000	1.0000000	1.0000000	NULL	NULL	NULL	NULL	NULL	NULL	2020-05-12 18:03:16.140
5	TEB30B20	2020-01-03 11:45:00.000	NULL	NULL	NULL	1.0000000	1.0000000	1.0000000	NULL	NULL	NULL	NULL	NULL	NULL	2020-05-12 18:03:16.140
6	TEB30B20	2020-01-03 11:15:00.000	NULL	NULL	NULL	1.0000000	1.0000000	1.0000000	NULL	NULL	NULL	NULL	NULL	NULL	2020-05-12 18:03:16.140
7	TEB30B20	2020-01-03 11:00:00.000	NULL	NULL	NULL	1.0000000	1.0000000	1.0000000	NULL	NULL	NULL	NULL	NULL	NULL	2020-05-12 18:03:16.140
8	TEB30B20	2020-01-03 10:45:00.000	NULL	NULL	NULL	1.0000000	1.0000000	1.0000000	NULL	NULL	NULL	NULL	NULL	NULL	2020-05-12 18:03:16.140
9	TEB30B20	2020-01-03 10:30:00.000	NULL	NULL	NULL	1.0000000	1.0000000	1.0000000	NULL	NULL	NULL	NULL	NULL	NULL	2020-05-12 18:03:16.140
10	TEB30B20	2020-01-03 10:15:00.000	NULL	NULL	NULL	1.0000000	1.0000000	1.0000000	NULL	NULL	NULL	NULL	NULL	NULL	2020-05-12 18:03:16.140
11	TEB30B20	2020-01-03 10:00:00.000	NULL	NULL	NULL	1.0000000	1.0000000	1.0000000	NULL	NULL	NULL	NULL	NULL	NULL	2020-05-12 18:03:16.140

- Barras con todos los valores perdidos: No es posible realizar un análisis nodal, al equipo de CHEC ya se le suministro un documento con aquellos circuitos que presentan falencias para este apartado.
- Barras donde no sea posible identificar el signo de las potencias para mas de un circuito, puesto que seria definir de manera arbitraria tanto el flujo como la magnitud de las variables del circuito, para completar un análisis nodal, además que entraríamos nuevamente al uso de combinatorias, que nuevamente estaría sujeto a un aumento considerable en el costo computacional.

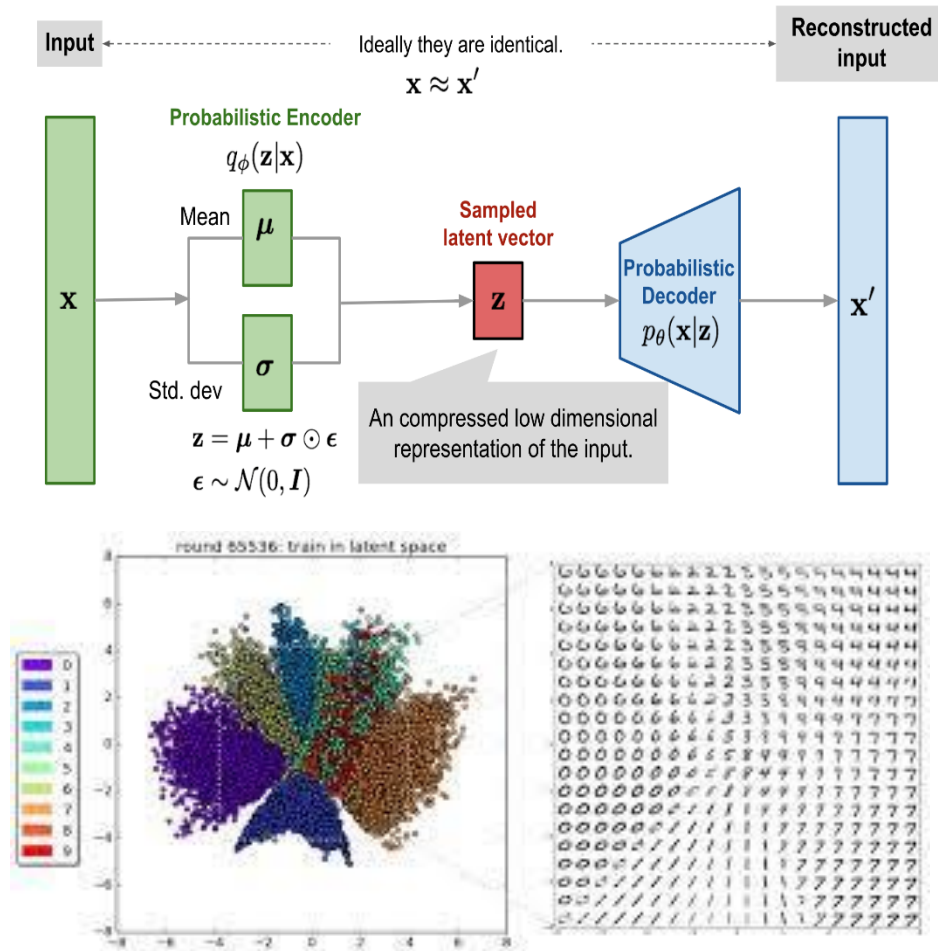
	CIRCUITO	TIEMPO_AJUSTADO	IA	IB	IC	VA	VB	VC	P	Q
1	IRR40B20	2020-01-01 00:00:00.000	NULL	NULL	NULL	120.5470000	120.5470000	118.8250000	NULL	NULL
2	IRR40L15	2020-01-01 00:00:00.000	48.0000000	46.0000000	46.0000000	121.1930000	120.7630000	119.9020000	-2.9600000	-11.2600000
3	IRR40L16	2020-01-01 00:00:00.000	NULL	0.0000000	0.0000000	NULL	NULL	NULL	0.0000000	0.0000000
4	IRR40L18	2020-01-01 00:00:00.000	17.0000000	18.0000000	16.0000000	119.3019000	NULL	NULL	0.9340000	3.8920000
5	IRR40L19	2020-01-01 00:00:00.000	27.0000000	29.0000000	29.0000000	123.5610000	123.3460000	122.4850000	6.1420000	0.8790000
6	IRR40T17	2020-01-01 00:00:00.000	24.0000000	0.0000000	18.0000000	NULL	NULL	NULL	-3.8600000	5.6100000
7	IRR40B20	2020-01-01 00:15:00.000	NULL	NULL	NULL	120.3320000	120.5470000	118.8250000	NULL	NULL
8	IRR40L15	2020-01-01 00:15:00.000	48.0000000	47.0000000	46.0000000	120.9780000	120.7630000	119.6860000	-2.7200000	-11.2600000
9	IRR40L16	2020-01-01 00:15:00.000	NULL	0.0000000	0.0000000	NULL	NULL	NULL	NULL	0.0000000
10	IRR40L18	2020-01-01 00:15:00.000	17.0000000	18.0000000	16.0000000	119.0092000	NULL	NULL	0.8250000	3.8740000

#### 4. Dificultades presentadas.

Presentamos un error en cuanto a la conexión continua a la base de datos, la idea sería saber si esto puede ser porque se presentó una coincidencia con un periodo de mantenimiento del sistema, o ya es cuestión de seguridad de chec que limita el número de peticiones o interrumpe la comunicación a ciertas horas del día o después de cierto periodo de tiempo haciendo consultas repetitivamente, en tal caso esta limitación debe ser resuelta.

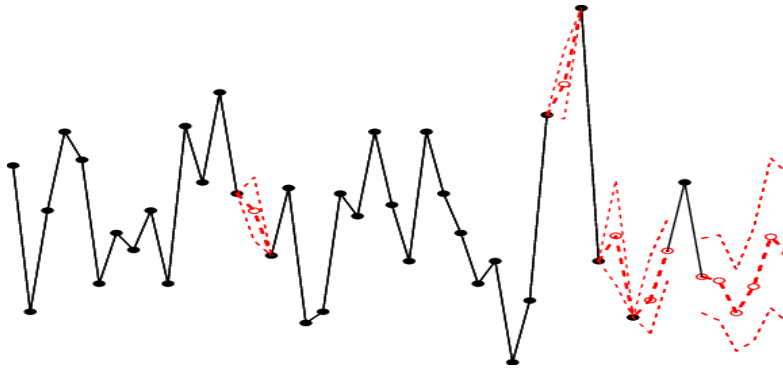


AUTOENCODERS, ya que precisamente esta es la ventaja que presentan, tienen la capacidad de aprender a definir un diccionario para mis datos de entrada, y en caso de que no sea conocido, o sea un circuito nuevo dado el caso, definirle una codificación diferente pero basándose en sistemas ya definidos, debido a codificación que se genera en el espacio latente.



- Los modelos de autoencoders, son las estructuras idóneas para la detección de datos atípicos, puesto que su entrenamiento se basa en lograr codificar series de tiempo con comportamientos adecuados, y en caso de que se presente una anomalía el mismo se encarga de depurarla en caso de no ser acorde con lo que el mismo sistema conoce, entonces no solo serviría como modelo de imputación sino como rectificador.





- Al manejar series de tiempo multivariadas, el modelo a utilizar no tiene que perder tanto la relación temporal como espacial, debido a esto el autoencoder, al necesitar estructurar en su salida las mismas dimensiones de entrada, se obliga a codificar también esas dos relaciones, evitando la pérdida en la reconstrucción y rectificación de las señales.

Aclaraciones del modelo:

- El planteamiento del modelo esta definido para el uso e imputación de las seis variables que corresponden a corrientes y voltajes en sus respectivas fases, debido a que tiene mayor certeza un cálculo directo a partir de voltajes y corrientes en el caso de las potencias.

## 6. Propuestas sprint 5.

- Pasar todos los registros de los circuitos de los sistemas SCADA y ABB desde el año 2020 al 2023, por el algoritmo de Análisis nodal; con el fin de obtener los datos con los cuales se entrenará el VAE.
- Definir la arquitectura del VAE a utilizar.
- Entrenar VAE para la reconstrucción de datos perdidos.

## 7. Sugerencia.

- En caso de que para este SPRINT no se entregue la base de datos del PRIME, se sugiere ya no tener en cuenta esta fuente de información para imputar datos en el sistema SCADA; ya que en el caso en el cual se llegue a utilizar esta fuente de información posteriormente, tocaría volver a entrenar el modelo de Deep learning incluyendo los datos imputados a partir del sistema PRIME, lo cual podría causar retrasos en el cronograma del proyecto.

