

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Tema:

**Evaluación oportuna de la vulnerabilidad dinámica de la red
usando técnicas de aprendizaje automático aplicado al
caso de estudio peruano**

Previo la obtención del Título de:

**MAGÍSTER EN ELECTRICIDAD, MENCIÓN SISTEMAS
ELÉCTRICAS DE POTENCIA**

Presentado por:

LUIS ANGEL HUARATAZO HUALLPA

GUAYAQUIL - ECUADOR

Año: 2025

DEDICATORIA

A mi querida esposa, Corina de la Cruz, gracias siempre por su apoyo aun en medio de toda situación.

A mis profesores de la maestría, cuyas enseñanzas y experiencias me ayudaron a desarrollar este trabajo.

Al Dr. Jaime Cepeda, por su apoyo, paciencia, conocimientos y sugerencias para la realización del presente trabajo de titulación.

Y en especial a nuestro Todopoderoso Dios, quien me supo fortalecer en medio de todas las dificultades que pase en este tiempo.

Infinitamente agradecido.

AGRADECIMIENTOS

Agradezco a los profesores de la ESPOL, entre ellos el Dr. Jaime Cepeda, por el tiempo que han dedicado para ayudarme en este trabajo.

A Raúl Levano, que me oriento en varios conceptos que necesitaba.

A mi esposa Corina de la Cruz, por estar recordándome que uno siempre es capaz de hacer lo que se ha propuesto.

Y nuevamente a Dios, por su amor y paciencia conmigo todo este tiempo.
Gracias.

DECLARACIÓN EXPRESA

Yo, Luis Angel Huaratazo Huallpa, acuerdo y reconozco que: La titularidad de los derechos patrimoniales de autor (derechos de autor) del proyecto de graduación corresponderá al autor, sin perjuicio de lo cual la ESPOL recibe en este acto una licencia gratuita de plazo indefinido para el uso no comercial y comercial de la obra con facultad de sublicenciar, incluyendo la autorización para su divulgación, así como para la creación y uso de obras derivadas. En el caso de usos comerciales se respetará el porcentaje de participación en beneficios que corresponda a favor del autor o autores. El estudiante deberá procurar en cualquier caso de cesión de sus derechos patrimoniales incluir una cláusula en la cesión que proteja la vigencia de la licencia aquí concedida a la ESPOL.

La titularidad total y exclusiva sobre los derechos patrimoniales de patente de invención, modelo de utilidad, diseño industrial, secreto industrial, secreto empresarial, derechos patrimoniales de autor sobre software o información no divulgada que corresponda o pueda corresponder respecto de cualquier investigación, desarrollo tecnológico o invención realizada por mí durante el desarrollo del proyecto de graduación, pertenecerán de forma total, exclusiva e indivisible a la ESPOL, sin perjuicio del porcentaje que me corresponda de los beneficios económicos que la ESPOL reciba por la explotación de mi innovación, de ser el caso.

En los casos donde la Oficina de Transferencia de Resultados de Investigación (OTRI) de la ESPOL comunique al autor que existe una innovación potencialmente patentable sobre los resultados del proyecto de graduación, no se realizará publicación o divulgación alguna, sin la autorización expresa y previa de la ESPOL.

Guayaquil, 11 de febrero del 2025.

Luis Angel Huaratazo Huallpa

EVALUADORES

Danny Ochoa Correa, PhD

Profesor de Materia

Jaime C. Cepeda Campaña, PhD

Tutor de Proyecto

RESUMEN

Este trabajo propone el desarrollo de la predicción de vulnerabilidad del sistema peruano a través del análisis de estabilidad oscilatoria, utilizando elementos de aprendizaje en base a data obtenida de diferentes condiciones de emergencia y/o contingencia del sistema peruano. Si bien ya existen herramientas de software para evaluar dichas contingencias, la idea de predecir con antelación condiciones de vulnerabilidad para ciertas condiciones de un nuevo escenario basado en aprendizaje maquina es una propuesta relativamente nueva.

Para ello este trabajo se apoya en herramientas como el análisis de contingencias para generación de data de escenarios aplicados a la red peruana. Además, mediante algunas técnicas de inteligencia artificial y otras de reducción de dimensionalidad, se clasifica los escenarios y entrena algoritmos para obtener con suficiente precisión estados del sistema que podrían ser vulnerables.

Se busca la predicción de estados para escenarios de prueba probables en base a cambios en los niveles de amortiguamiento de la red. Además, se hace notar que el tiempo de procesamiento e identificación de condiciones mediante un modelo bien entrenado de IA es reducido, añadiéndole importancia.

Del presente trabajo se desprende que, con un buen entrenamiento y data necesaria, se puede predecir estados vulnerables en el sistema peruano que den pie a la alerta temprana. Además, que algoritmos entrenados ayudan en la identificación en tiempo real. También que no importando la naturaleza de la red (en este caso se hizo uso de la red actualizada con proyectos renovables incluidos), y con data de cierta cantidad de parámetros elegidos se puede identificar condiciones de probable vulnerabilidad.

En decir, siendo muy factible la evaluación rápida de la estabilidad oscilatoria del sistema peruano mediante la clasificación de la data, se puede aplicar sin problemas a la data Scada – WAMS del sistema peruano.

Palabras clave: Minería de datos, entrenamiento, redes, aprendizaje automático, etc.

ABSTRACT

This work proposes the development of the vulnerability prediction of the Peruvian system through the analysis of oscillatory stability, using learning elements based on data obtained from different emergency and/or contingency conditions of the Peruvian system. Although software tools already exist to evaluate such contingencies, the idea of predicting vulnerability conditions in advance for certain conditions of a new scenario based on machine learning is a relatively new proposal.

To do this, this work relies on tools such as contingency analysis to generate scenario data applied to the Peruvian network. Furthermore, using some artificial intelligence techniques and other dimensionality reduction techniques, the scenarios are classified and algorithms are trained to obtain system states that could be vulnerable with sufficient precision.

State prediction is sought for probable test scenarios based on changes in network buffer levels. In addition, it is noted that the processing time and identification of conditions using a well-trained AI model is reduced, adding importance.

From this work it is clear that, with good training and necessary data, vulnerable states can be predicted in the Peruvian system that give rise to early warning. Furthermore, trained algorithms help in real-time identification. Also, regardless of the nature of the network (in this case the updated network was used with renewable projects included), and with data from a certain number of chosen parameters, conditions of probable vulnerability can be identified.

In other words, the rapid evaluation of the oscillatory stability of the Peruvian system by classifying the data is very feasible, and it can be applied without problems to the Scada – WAMS data of the Peruvian system.

Keywords: Data mining, training, network, machine learning, etc.

ÍNDICE GENERAL

EVALUADORES	5
RESUMEN	I
<i>ABSTRACT</i>	II
ÍNDICE GENERAL.....	III
ABREVIATURAS	VI
SIMBOLOGÍA	VII
ÍNDICE DE FIGURAS.....	VIII
ÍNDICE DE TABLAS	X
CAPÍTULO 1	1
1. Introducción.....	1
1.1 Descripción del problema	2
1.2 Justificación del problema.....	2
1.3 Objetivos.....	2
1.3.1 Objetivo General	2
1.3.2 Objetivos Específicos	2
1.4 Estado del Arte	3
1.4.1 Eventos en Perú.....	3
1.4.2 Identificación de vulnerabilidad en sistemas de potencia.....	4
1.5 Marco Teórico	6
1.5.1 Red Eléctrica Peruana	6
1.5.2 Estabilidad de Sistemas de Potencia	8
1.5.3 Evaluación de la vulnerabilidad	9
1.5.3.1 Métodos Fuera de Línea.....	10
1.5.3.2 Métodos en Línea	10

1.5.3.3	Métodos en Tiempo Real.....	10
1.5.4	PowerFactory de DlgSILENT	11
1.5.5	Python	12
1.5.6	Minería de datos.....	13
1.5.6.1	Scikit Learn.....	14
1.5.6.2	Random Forest.....	15
1.5.6.3	Logistic Regression	16
1.5.6.4	K Nearest Neighbours	16
1.5.6.5	Resampling.....	17
1.5.6.6	Escalamiento de datos	18
1.5.6.7	Matriz de confusión.....	19
1.5.6.8	Cross Validation.....	20
1.5.6.9	Análisis componentes principales.....	21
CAPÍTULO 2.....		22
2.	Metodología	22
2.1	Reducción de red y generación de data del sistema peruano	25
2.2	Proceso de entrenamiento de data.....	28
2.2.1	Escalamiento, reducción de dimensionalidad y cross validation	29
2.2.2	División de los datos	31
2.2.3	Predicción mediante modelos y comparación	32
2.3	Viabilidad técnica	33
CAPÍTULO 3.....		34
3.	Resultados y análisis.....	34
3.1	Preparación de la red.....	34
3.1.1	Reducción del Sistema peruano	34
3.1.2	Elección de elementos del sistema eléctrico	35

3.2	Generación de datos.....	38
3.3	Procesamiento de datos:	40
3.3.1	Resampling y entrenamiento de datos:	40
3.3.2	Reducción por componentes principales:.....	43
3.3.3	Entrenamiento y evaluación del modelo:.....	43
3.3.4	Viabilidad técnica – económica	46
CAPÍTULO 4		49
4.	CONCLUSIONES Y RECOMENDACIONES	49
4.1	CONCLUSIONES:	49
4.2	RECOMENDACIONES:.....	50
BIBLIOGRAFÍA		51

ABREVIATURAS

ESPOL Escuela Superior Politécnica del Litoral

ASTM American Society for Testing and Materials

IEEE Instituto de Ingenieros Eléctricos y Electrónicos

SIMBOLOGÍA

m	Metro
V	Voltio
I	Corriente
P	Potencia
ζ	Amortiguamiento

ÍNDICE DE FIGURAS

Figura 1.1 DVRs determinadas para el sistema New England.	5
Figura 1.2 Mapa unifilar del SEIN de Perú.....	6
Figura 1.3 <i>Demanda media horaria ejecutada y producción acumulada del sistema Peruano [9].</i>	7
Figura 1.4 <i>Clasificación actual de la estabilidad del sistema de Potencia.</i>	8
Figura 1.5 Métodos de evaluación de vulnerabilidad del sistema de Potencia [8].	11
Figura 1.6 Software Power Factory de Digsilent.	12
Figura 1.7 Python sobre la aplicación web Open Source Jupyter Notebook.	13
Figura 1.8 Concepto escalonado sobre la minería de datos.....	14
Figura 1.9 Conceptualización sobre Scikit Learn.	15
Figura 1.10 Conceptualización sobre Random Forest.	15
Figura 1.11 Separación de data producto de una regresión logística.	16
Figura 1.12 Conceptualización sobre K-Nearest Neighbors	17
Figura 1.13 Ejemplificación del sub muestreo y sobre muestreo.	18
Figura 1.14 Mejora en el tratamiento de datos cuando se usa escalamiento.	19
Figura 1.15 Matriz de confusión.....	20
Figura 1.16 Alternación característica en la data propio del Cross Validation.	21
Figura 1.17 Descomposición en dimensiones luego de aplicar PCA.	21
Figura 2.1 Esboce gráfico de la metodología planteada.	22
Figura 2.2 Metodología de Identificación de vulnerabilidad de la red peruana.	23
Figura 2.3 Diagrama de flujo para el scripting de PowerFactory.....	24
Figura 2.4 Herramienta para reducción de redes – Network Reduction	25
Figura 2.5. Boundaries o fronteras realizadas para la reducción de red.....	26
Figura 2.6 Descripción de fronteras para reducciones.....	26
Figura 2.7 Líneas y barajes seleccionados.....	27
Figura 2.8 Scripting – PowerFactory.....	27
Figura 2.9 Diagrama de flujo para el scripting de PowerFactory.....	29
Figura 2.10 Método del codo.	30

Figura 2.11 Cantidad de componentes principales a elegir mediante método de codo.	30
Figura 2.12 Técnica del cross validation.....	31
Figura 2.13 Técnica del cross validation aplicado a los datos generados.	31
Figura 2.14 Ejemplo de división de cierta data.	32
Figura 2.15 Comparación de modelos de entrenamiento.	32
Figura 2.16 Elementos necesarios para implementación de algoritmos IA aplicados.	
.....	33
Figura 3.1 Red representativa del sistema eléctrico peruano.	34
Figura 3.2 Áreas a reducir.	35
Figura 3.3 Red reducida.	35
Figura 3.4 Trazos de los elementos de red representativos del sistema eléctrico peruano.....	36
Figura 3.5 Script desarrollado en Python para el análisis.	38
Figura 3.6 Criterios relacionados a eigenvalores para captura de parámetros.	39
Figura 3.7 Patrón de comportamiento de los amortiguamientos mínimos.	40
Figura 3.8 Distribución de contingencias que superan o no el criterio.	41
Figura 3.9 Remuestreo de los datos obtenidos.	41
Figura 3.10 Data de entrenamiento previo al escalamiento.	42
Figura 3.11 Data de entrenamiento después del escalamiento.	42
Figura 3.12 Método del codo aplicado a la data obtenida.	43
Figura 3.13 División de la data entrenamiento, prueba y validación.	44
Figura 3.14 Comparación entre los modelos aplicados.	44
Figura 3.15 Elección de parámetros de optimización.	45
Figura 3.16 Matriz de confusión obtenida.	45
Figura 3.17 Estructura física para la evaluación de vulnerabilidad del sistema.	47

ÍNDICE DE TABLAS

Tabla 1.1 Estructura de la data generada para entrenamiento.....	28
Tabla 1.2 Lineas de transmisión en análisis.	36
Tabla 1.3 Barras de transmisión en análisis	37
Tabla 1.4 Data Generada.	39
Tabla 1.2 Media y desviación de los datos.	44
Tabla 1.2 Esboce de costes asociados.....	47

CAPÍTULO 1

1. INTRODUCCIÓN

El objetivo de los sistemas eléctricos de potencia, desde hace muchos años, ha sido la distribución eficiente de electricidad, lo cual involucra la continuidad, seguridad y calidad del servicio. En base a esto, los sistemas deben ir más allá de solo proteger y despejar fallas cuando estas se presenten, sino que es necesario tener herramientas que tengan la capacidad de dar cuenta de ocurrencia de fenómenos, las cuales deben de estar ya embebidas en nuestros sistemas y trabajando en tiempo real con adecuadas filosofías de trabajo que apunten a cumplir las exigencias mencionadas.

Una de estas herramientas que hoy en día van encontrando gran aplicación en las redes eléctricas son las técnicas de minería de datos e Inteligencia artificial.

Estas, debidamente entrenadas, nos proporcionan rapidez a la operación además de detección de probables condiciones vulnerables, que en otro caso (procesamiento analítico), demorarían varios minutos inclusive hasta horas.

El presente trabajo utiliza código computacional para entrenar algoritmos de aprendizaje supervisado del tipo de clasificación con fin determinar, en base a valores de variables de sistema ubicados en diferentes partes del sistema peruano, posibles condiciones vulnerables de estabilidad oscilatoria.

Para ello se usará cálculos de valores de amortiguamiento para cada escenario, (se irá variando los escenarios) y guardando la data para entrenamiento. Luego se entrenará diferentes algoritmos de Machine Learning (del tipo clasificación), y se hace una evaluación de comparación entre ellos, eligiendo el que mejor resultado nos brinde.

La gran cantidad de variables es reducida usando el análisis de componentes principales (PCA); el cual nos reduce el tiempo y la exigencia del procesamiento.

Una vez entrenado los algoritmos, estos se comparan para la data de prueba.

Todos estos algoritmos son elaborados usando el software Python (Jupyter Notebook).

1.1 Descripción del problema

Los cambios de escenario operativos del sistema podrían devenir en que este se vuelva inseguro o reduzca su margen de seguridad. Una exploración de las técnicas de modernas no encuentra aún aplicación difundida en las redes actuales en Perú.

Estas causales se reflejan en un desconocimiento y no adecuada preparación frente a estos fenómenos de parte de muchos operadores y por lo tanto, la capacidad de tener conciencia situacional para toma de decisiones se ve reducida.

Por lo que, el problema principal a afrontar consiste en que la demora en un diagnóstico del estatus de vulnerabilidad del sistema eléctrico y la ejecución del respectivo control sobre la red frente a una determinada contingencia, puede desencadenar un evento en cascada y un eventual colapso.

1.2 Justificación del problema

1.3 Objetivos

1.3.1 Objetivo General

- Realizar la evaluación oportuna de la vulnerabilidad dinámica del sistema eléctrico peruano mediante técnicas de aprendizaje automático para la mejora de tiempos de respuesta frente a fenómenos de red.

1.3.2 Objetivos Específicos

- Dar indicación del estatus de vulnerabilidad dinámica del sistema y aplicarlos al sistema eléctrico peruano.
- Desarrollar metodología para determinar la conciencia situacional de la red basadas en aprendizaje automático.

- Aplicar lo desarrollado a la evaluación de la vulnerabilidad del sistema eléctrico peruano.

1.4 Estado del Arte

1.4.1 Antecedentes en Perú

En Perú se han producido eventos relacionados a fenómenos de estabilidad oscilatoria en años pasados que llevaron a la salida de algunos elementos de sistema, como las desconexiones en la C.H. La Joya en 2018 o las líneas L-6652 (2017) y L-6645 (2018).

Estudios de conexión internacionales realizados con fin de interconexión del área norte se nos menciona que para determinados escenarios no ha sido posible la operatividad debido a que el amortiguamiento del nuevo sistema conjunto no es capaz de atenuar las oscilaciones que se presentaban. [1]

Estos inconvenientes causaban perdidas de generación para el caso de centrales, así como la imposibilidad de tener complementariedad hidrológica para el caso de interconexiones. Esto no ha permitido que la región norte del país pase a ser un sistema sólido, teniendo el problema de continuar con característica radial pese a que en los últimos años se han venido desarrollando algunos estudios y propuestas de proyectos para la construcción de líneas en 220 y 500 kV para su integración.[2]

En [3] se presenta un memorial técnico de la oscilación de potencia presentada entre las áreas centro y sur en el año 2013. Esta oscilación fue de gran notoriedad en el sistema peruano debido a la salida de las líneas L-2053/L-2054 entre las subestaciones Cotaruse y Socabaya; lo que produjo una desconexión de aproximadamente 215 MW, además de un tiempo de interrupción alto para la reposición (luego de aproximadamente media hora se logra unir nuevamente los sistemas Centro Norte y Sur).

Recientemente en 2023 se presentó un fenómeno de desconexión muy grande, con aproximadamente 841 MW perdidos [4], lo cual nos reitera la necesidad de herramientas que permitan dar cuenta de posibles colapsos o estados inseguros de la red.

1.4.2 Identificación de vulnerabilidad en sistemas de potencia

Los desafíos para identificación de condiciones vulnerables del sistema eléctrico han venido incrementándose en estos últimos años, debido principalmente a la inserción de las nuevas tecnologías de generación y aumento de la complejidad de la red.

La tradicional evaluación offline y posterior análisis de estos resultados mediante estudios de gabinete tiene los grandes inconvenientes de usar sistemas que no están debidamente modelados, además del largo tiempo de simulación que requiere.

Por ello, novedosas técnicas surgen para aplicación de herramientas de aprendizaje maquina y aprendizaje profundo. En [5] se recopila una selección de modelos de Machine Learning que podrían ser útiles para el diagnóstico de estados del sistema. Fenómenos como cortes de energía, evaluación y/o control de estabilidad y restauración del sistema son abordados.

Existen propuestas que usan enfoques de Machine Learning, pues toman como base datos históricos, parámetros de operatividad de la red y proponen determinados índices de vulnerabilidad que describe los aspectos de resiliencia y confiabilidad de los componentes de red. [6]

También existen propuestas de aplicación que consiste en la evaluación del sistema en base al amortiguamiento que presenta, además de considerar otros factores como la frecuencia, amortiguamiento y ángulos. En [7] se aprecia dicha evaluación sobre un área de la costa nor-oeste de los Estados Unidos para diferentes escenarios, siguiendo con un análisis Prony que permite obtener los amortiguamientos. Luego un análisis PCA y la técnica CART (Classification and

regression trees) permiten dar cuenta de los posibles estados vulnerables posibles.

En [8] se nos plantea una metodología muy interesante y bastante completa. Las medidas del sistema son recogidas mediante PMUs (información de entrada), seguidamente se busca estimar regiones de vulnerabilidad dinámica (DVR), las cuales estarán divididas en 2 áreas: la región no vulnerable y la vulnerable, las cuales se definen mediante un hiperplano.

Previamente se determina de forma aproximada la distribución espacial de las DVRs mediante un método de reconocimiento de patrones basado en Funciones Empíricas Ortogonales (EFO).

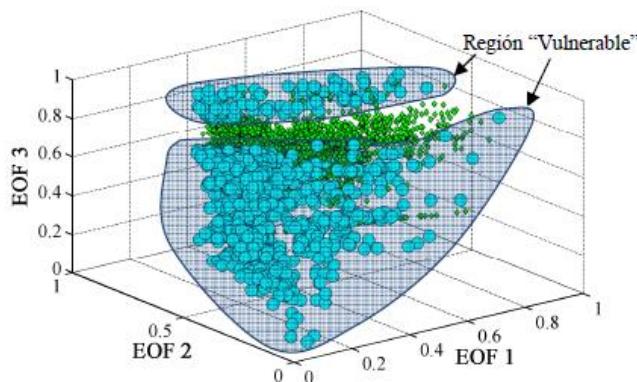


Figura 1.1 DVRs determinadas para el sistema New England.

Una vez determinadas las regiones, se ubican las posiciones del estado dinámico real mediante un clasificador inteligente (Support Vectorial Machine (SVC)), el cual se entrena previamente fuera de línea. El sistema entregara 1 para estados vulnerables y 0 para estados no vulnerables.

Por todo lo expuesto, la evaluación de la vulnerabilidad del sistema busca echar mano de las herramientas IA y que su uso de ellas para sistemas reales y complejos son necesarias.

1.5 Marco Teórico

1.5.1 Red Eléctrica Peruana

El Sistema Eléctrico Interconectado Nacional (SEIN) de Perú (figura 1.2) es el sistema eléctrico que conecta varias regiones del país, permitiendo el flujo de energía entre diferentes zonas y la confiabilidad del suministro.

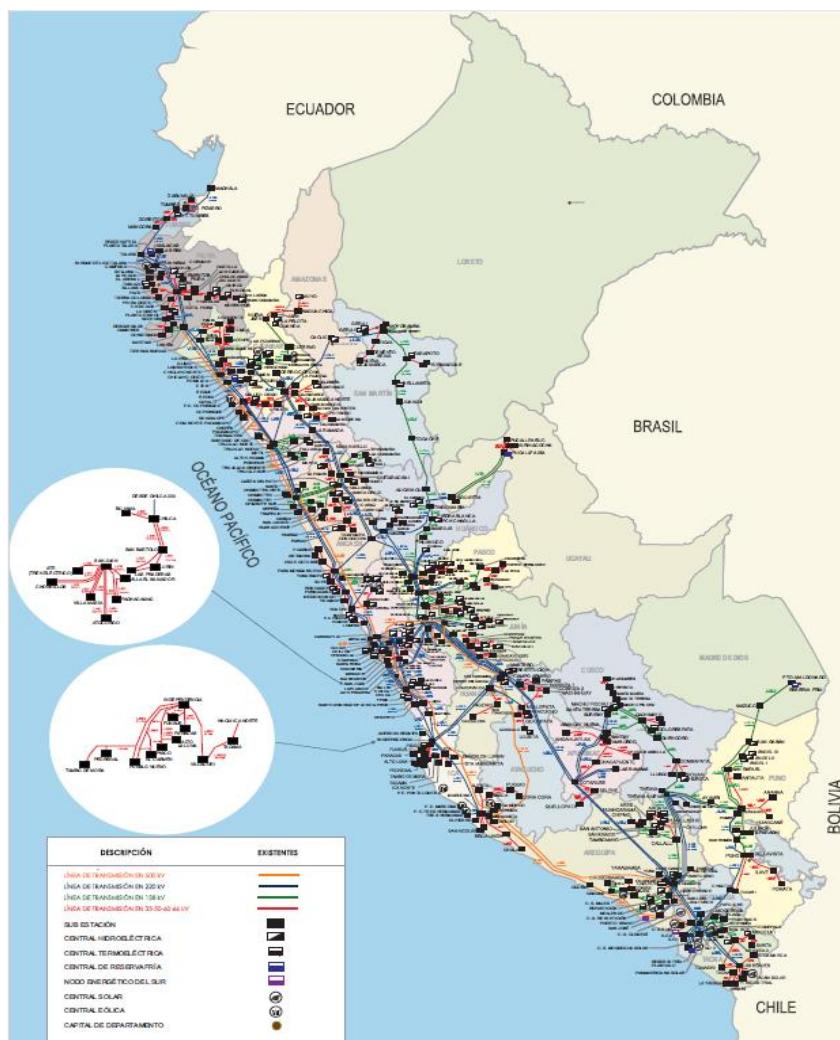


Figura 1.2 Mapa unifilar del SEIN de Perú.

Para ello utiliza una combinación de energías renovables (hidráulica, solar y eólica) y fuentes no renovables (gas natural y térmica) siendo la mayor parte de la generación

proveniente de centrales hidroeléctricas, especialmente en la sierra y selva, debido al potencial hídrico presente en esas regiones.

Anteriormente el sistema estaba dividido en áreas aisladas y susceptible a eventos, más las dos últimas décadas ha estado en constante expansión y modernización para atender la creciente demanda de electricidad y mejorar la eficiencia de la infraestructura. Esto ha permitido que las áreas Centro - Norte y Sur se interconecten, mejorando la robustez del sistema.

Aunque se han realizado los avances mencionados, todavía existen áreas rurales y remotas con limitado acceso a la electricidad, como los sistemas eléctricos aislados en regiones de la amazonía, que no están conectados al SEIN, lo que dificulta la integración total del país, generando desafíos en términos de inclusión energética.

La producción nacional se refleja en las siguientes figuras:

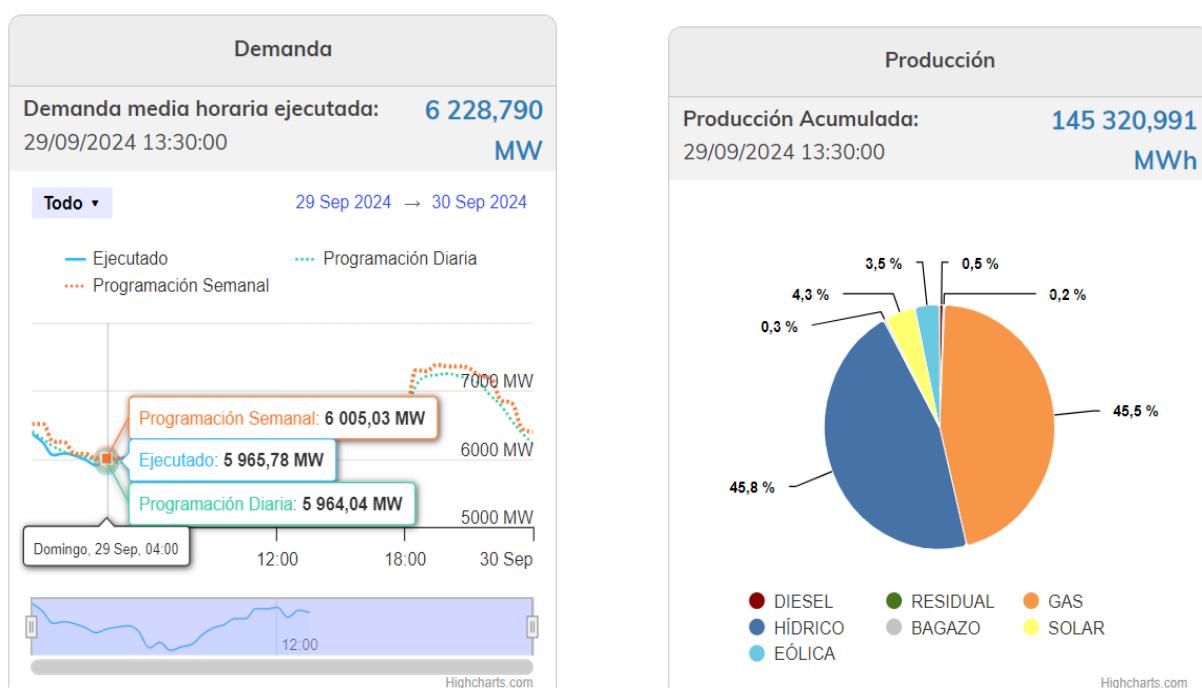


Figura 1.3 Demanda media horaria ejecutada y producción acumulada del sistema Peruano [9].

1.5.2 Estabilidad de Sistemas de Potencia

De acuerdo a [10] la estabilidad del sistema de potencia es la capacidad del sistema eléctrico, para una determinada condición de operación inicial, recuperar un estado de equilibrio operativo después de haber sido sometido a una perturbación física, con el resultado de las variables del sistema se encuentren limitadas de modo que prácticamente todo el sistema permanezca intacto.

La estabilidad de sistemas eléctricos nos propone retos a los cuales debemos enfrentar con fin de asegurar la seguridad del sistema. Con fin de abordar soluciones a dichos retos, es que se ha venido elaborando clasificaciones de la estabilidad: Una primera definición, que ahora es considerada clásica, evolucionó gradualmente debido a la penetración de nuevas tecnologías, por lo que, en 2016 la clasificación fue redefinida [10]. Esta moderna clasificación se muestra en la figura 1.4.



Figura 1.4 Clasificación actual de la estabilidad del sistema de Potencia.

La estabilidad del ángulo del rotor tiene que ver con la capacidad de las máquinas síncronas interconectadas en un sistema de potencia para permanecer en sincronismo en condiciones normales de operación y para recuperar el sincronismo después de ser sometido a una pequeña o grande perturbación.

La estabilidad de tensión se refiere a la capacidad de un sistema de potencia para mantener tensiones constantes cercanas al valor nominal en todas las barras del sistema después de haber estado sujeto a una perturbación.

La estabilidad de resonancia se explica a partir del fenómeno de resonancia. Esta, en general, ocurre cuando el intercambio de energía se realiza periódicamente de manera oscilatoria. Estas oscilaciones crecen en caso de disipación insuficiente de energía en la trayectoria del flujo y se manifiestan (en los sistemas de energía eléctrica) en la magnificación de las magnitudes de voltaje/corriente/torque. Cuando estas magnitudes superan los umbrales especificados, se dice que se ha producido una inestabilidad de resonancia.

La estabilidad de convertidores se basa en que la amplia escala de tiempo relacionada con los controles de los CIG puede dar lugar acoplamientos cruzados tanto con la dinámica electromecánica de las máquinas como con los transitorios electromagnéticos de la red, y puede conducir a oscilaciones inestables del sistema de potencia en un amplio rango de frecuencias.

La estabilidad de frecuencia se define como habilidad del sistema de mantener el balance generación – carga después de haber sido sometido a una perturbación. [11]

1.5.3 Evaluación de la vulnerabilidad

De acuerdo a [8] la vulnerabilidad es definida como “el nivel de riesgo presentado por un sistema eléctrico de potencia durante una condición de operación estática o dinámica determinada respecto de la ocurrencia de eventos en cascada”.

Este concepto le da a la vulnerabilidad la característica de ser un indicador esencial de la proximidad del sistema a un colapso. Por lo que, es notorio que la evaluación de la vulnerabilidad es una tarea fundamental dentro de la estructura de una red auto curable

ya que tiene la función de detectar la necesidad de llevar a cabo acciones de control de área extendida.

También en [8] se presenta una clasificación de las técnicas para evaluar vulnerabilidad basada en su potencial aplicación en los centros de control.

Así, los métodos pueden ser: fuera de línea, en línea, y en tiempo real.

1.5.3.1 Métodos Fuera de Línea

La evaluación fuera de línea se realiza mediante métodos convencionales basados en simulaciones de alta complejidad que generalmente conforman tareas con gran consumo de tiempo computacional, lo que restringe las aplicaciones en línea [8].

1.5.3.2 Métodos en Línea

Los datos de entrada se actualizan a través de equipos de medición y herramientas de procesamiento adecuadas (SCADA/EMS, PMU, GIS) en la evaluación en línea, pero los resultados no necesariamente se obtienen tan rápido como ocurre la evolución de los eventos en tiempo real [8].

1.5.3.3 Métodos en Tiempo Real

Los datos de entrada reflejan la imagen más reciente de las condiciones del sistema en un análisis en tiempo real, y todo el proceso se lleva a cabo dentro de muy poco tiempo, por lo general no superior a un par de segundos [8].

En la siguiente figura se puede apreciar dicha clasificación:

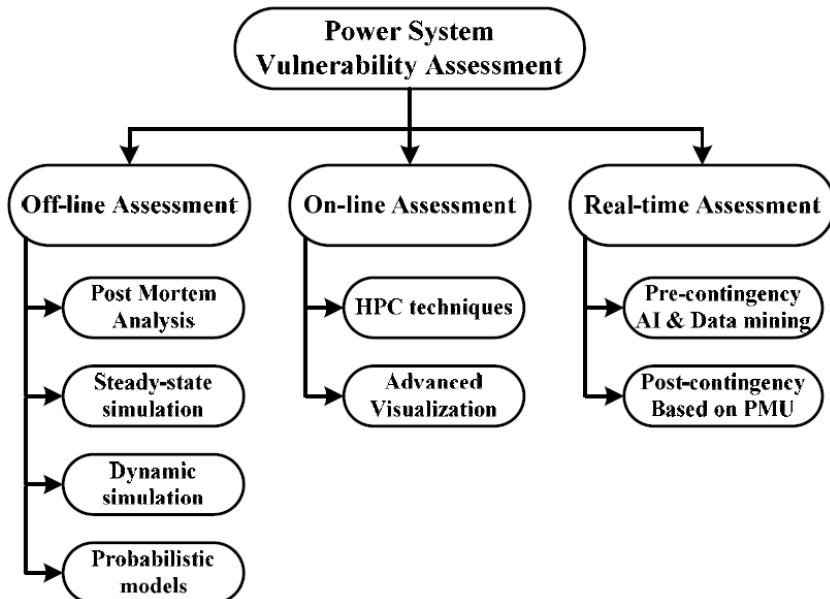


Figura 1.5 Métodos de evaluación de vulnerabilidad del sistema de Potencia [8].

1.5.4 PowerFactory de DIgSILENT

El programa DlgSILENT PowerFactory, es una herramienta de ingeniería asistida por ordenador para el análisis de transmisión, distribución y sistemas de energía eléctrica industrial. Ha sido diseñado como un paquete de software avanzado integrado e interactivo dedicado al sistema y control de energía eléctrica con fin de alcanzar los principales objetivos de planificación y optimización de la operación [12].

Para el presente trabajo usaremos los módulos de simulación RMS, flujo de carga y scripting.

Las redes eléctricas a utilizar serán analizadas mediante el software para los estados estacionario y transitorio (electromecánico).

Para la generación de data para usar en los modelos de entrenamiento, utilizaremos la herramienta de scripting que maneja el software.

Se presenta a continuación una vista de la interfaz del software:

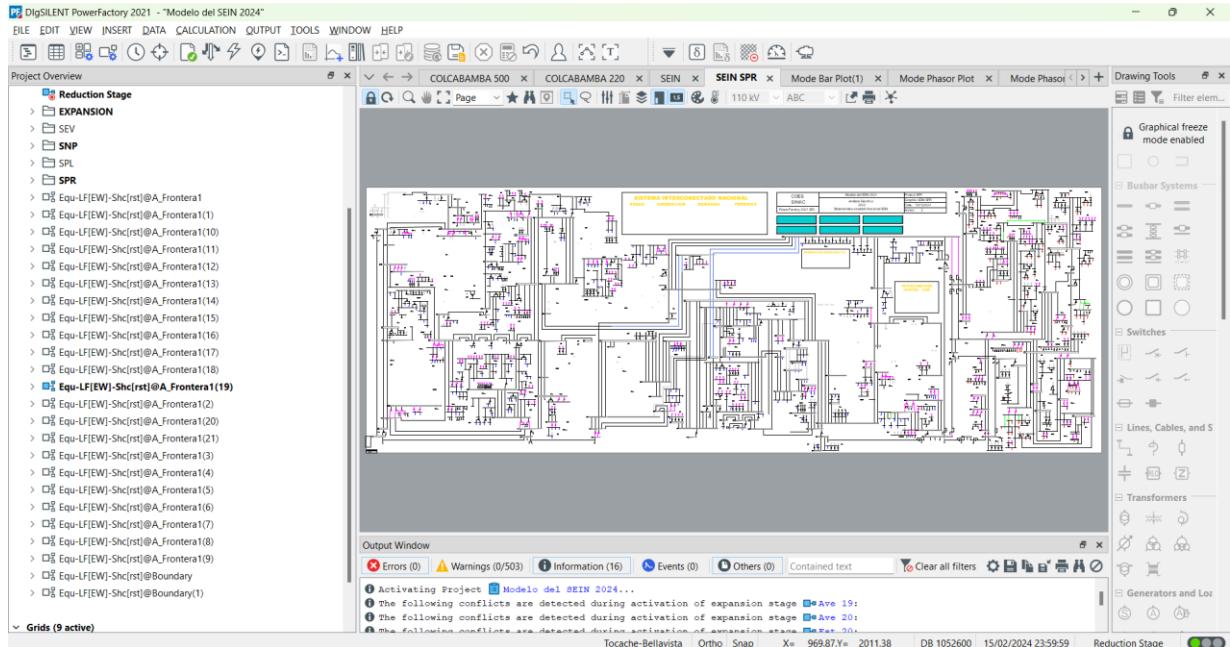


Figura 1.6 Software PowerFactory de Digsilent.

1.5.5 Python

Python se caracteriza por ser un lenguaje de programación de alto nivel, que tiene una sintaxis sencilla y legible.

También se caracteriza por ser un lenguaje interpretado, lo que significa que no es necesario compilar el código antes de ejecutarlo. A su vez, como Python se ejecuta línea por línea, al detectarse un error en alguna de ellas, el programa se detiene y resalta dónde se encuentra la línea que el programador debe corregir. Además es de programación multiplataforma, lo que significa que el mismo código puede ser ejecutado en diferentes sistemas operativos sin necesidad de realizar modificaciones.

Python es utilizado para darle vida a una amplia variedad de aplicaciones, desde automatizaciones hasta el machine learning. También es uno de los lenguajes más populares para interactuar con bases de datos y acceder a ellas de forma segura, optimizando el tiempo de respuesta de las solicitudes de información y facilitando el procesamiento de los datos.

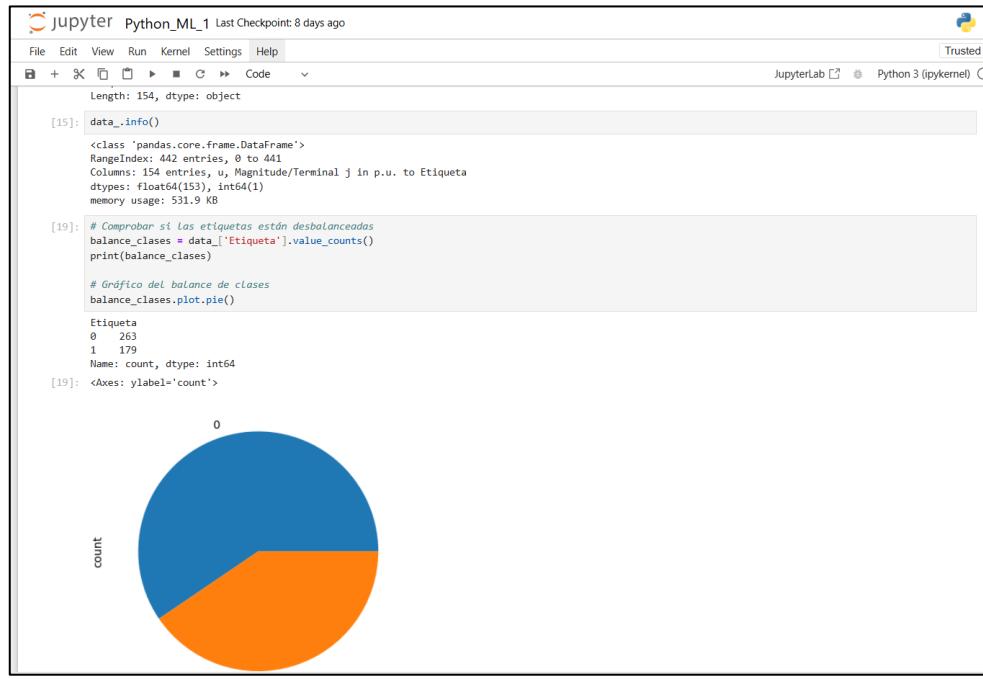


Figura 1.7 Python sobre la aplicación web Open Source Jupyter Notebook.

En la figura 1.7 se presenta la interfaz de fuente libre que trabaja sobre la web, Jupyter Notebook, el cual es soportado por la interfaz gráfica de usuario Anaconda Navigator, mostrando código de aprendizaje automático.

1.5.6 Minería de datos

La minería de datos es el proceso de hallar anomalías, patrones y correlaciones en grandes conjuntos de datos con fin de predecir resultados empleando una amplia variedad de técnicas. Se puede utilizar esta información para incrementar beneficios, recortar costos técnicos, computacionales, reducir riesgos y más.

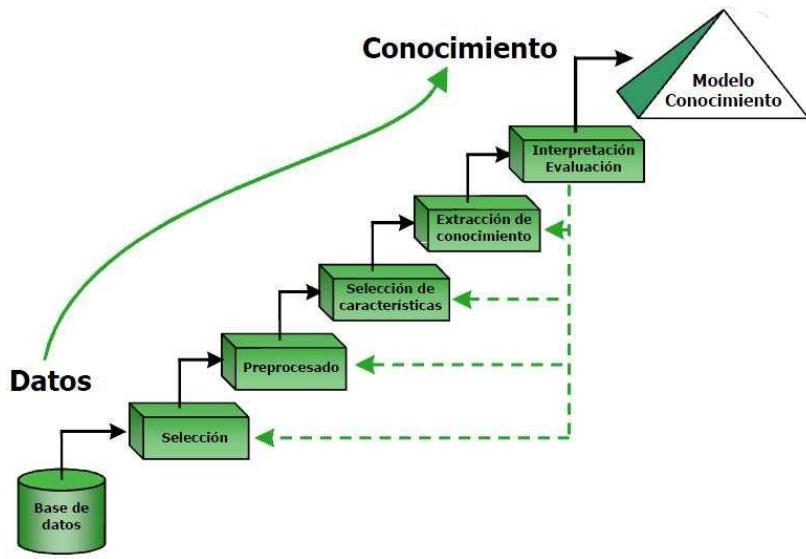


Figura 1.8 Concepto escalonado sobre la minería de datos.

1.5.6.1 Scikit Learn

Scikit-learn es una biblioteca de machine learning en Python, diseñada para simplificar la implementación de algoritmos de aprendizaje automático en diversas aplicaciones. Se construye sobre bibliotecas populares como NumPy, SciPy y Matplotlib, permitiendo a los desarrolladores entrenar, evaluar y ajustar modelos de aprendizaje automático sin una programación compleja.

Sus principales características incluyen herramientas como limpieza y preparación de datos, clasificación, regresión, clustering, reducción de dimensionalidad, selección de modelo, validación cruzada, métricas de rendimiento, preprocesamiento.

Es una de las bibliotecas de machine learning más populares y se usa ampliamente en investigación y en aplicaciones técnico - empresariales.

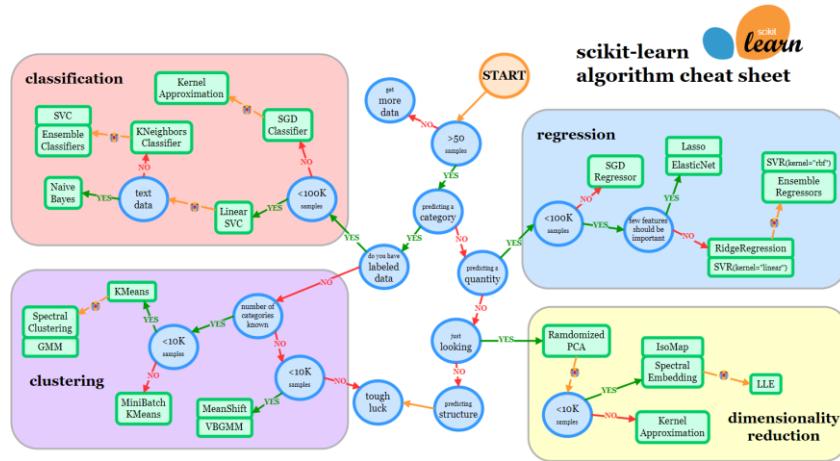


Figura 1.9 Conceptualización sobre Scikit Learn.

1.5.6.2 Random Forest

Random Forest es un algoritmo de machine learning basado en árboles de decisión que combina múltiples árboles para mejorar la precisión y reducir el riesgo de sobreajuste (overfitting). Fue introducido por Leo Breiman en 2001 y es conocido por su versatilidad en problemas de clasificación y regresión, así como por su capacidad para manejar grandes conjuntos de datos y características desbalanceadas.

Random Forest es un ejemplo de un método de ensamble, en el cual varios modelos "débiles" (en este caso, árboles de decisión) trabajan juntos para producir un modelo "fuerte".

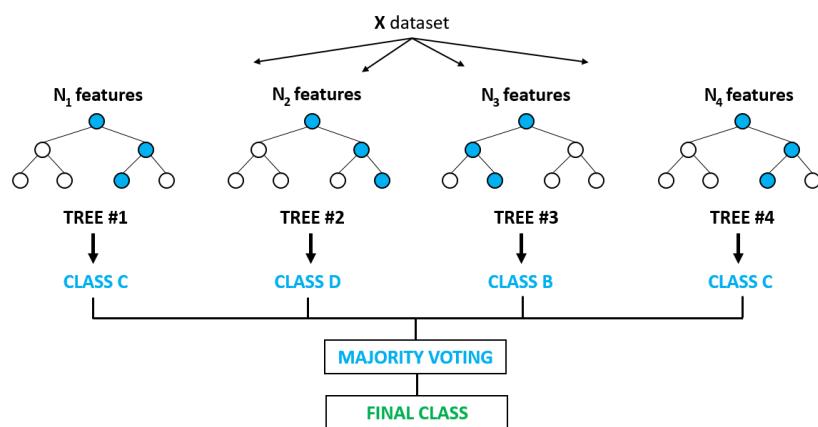


Figura 1.10 Conceptualización sobre Random Forest.

1.5.6.3 Logistic Regression

La regresión logística es un modelo estadístico que se utiliza para predecir la probabilidad de una clase o evento, como la presencia o ausencia de una característica, basado en uno o más predictores (características). Es especialmente útil para problemas de clasificación binaria, aunque también puede extenderse a casos multiclase.

Su simplicidad, eficiencia e interpretabilidad la convierten en una opción popular, aunque es importante considerar su suposición de linealidad y la necesidad de transformaciones cuando se manejan relaciones más complejas en los datos.

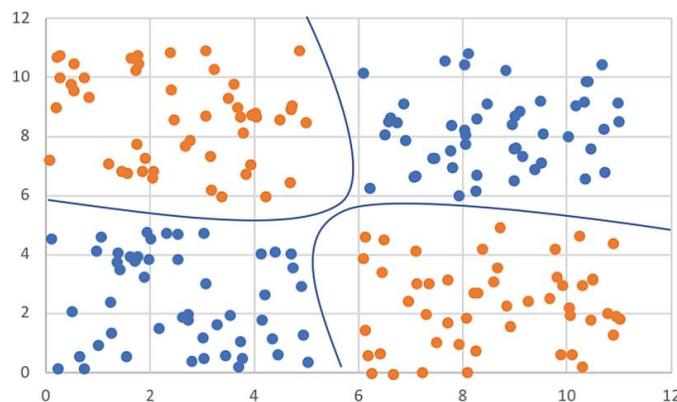


Figura 1.11 Separación de data producto de una regresión logisitica.

1.5.6.4 K Nearest Neighbours

K-Nearest Neighbors (KNN) es un algoritmo de aprendizaje supervisado utilizado principalmente para problemas de clasificación, aunque también se puede utilizar para regresión. Se basa en la idea de que los puntos de datos similares estarán cerca unos de otros en el espacio de características. KNN es un método simple y efectivo, especialmente para conjuntos de datos pequeños y cuando no se requiere una modelación compleja.

Es un algoritmo poderoso y fácil de usar, especialmente para problemas donde la relación entre los datos es difícil de modelar. Aunque tiene algunas limitaciones, su

versatilidad y efectividad en muchos contextos lo convierten en una buena opción para tareas de clasificación y regresión.

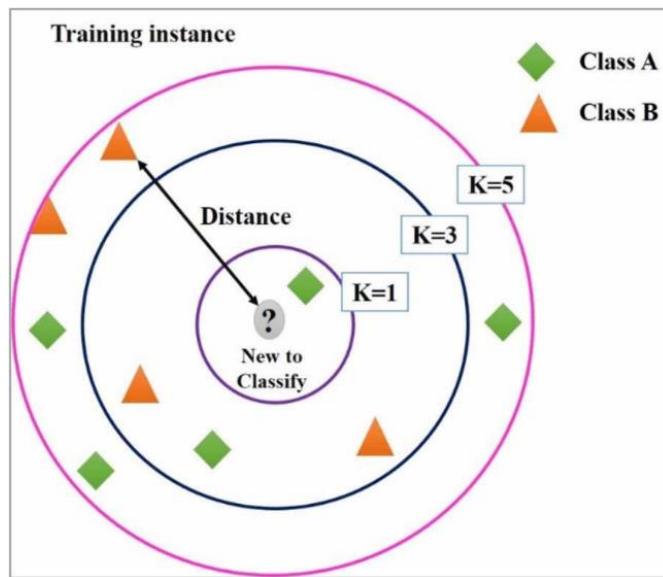


Figura 1.12 Conceptualización sobre K-Nearest Neighbors

1.5.6.5 Resampling

El balanceo de datos o Resampling es una técnica importante en machine learning, especialmente en problemas de clasificación donde las clases están desbalanceadas. Esto ocurre cuando una clase tiene significativamente más muestras que otra, lo que puede llevar al modelo a "aprender" a predecir la clase mayoritaria para minimizar el error, sacrificando la precisión en la clase minoritaria.

Entre los métodos tenemos:

Recolección de más datos:

Obtener más muestras de la clase minoritaria es la solución ideal, pero suele ser difícil o costoso.

Submuestreo (Undersampling):

Se reduce el tamaño de la clase mayoritaria para que tenga un número similar al de la clase minoritaria, aunque esto puede descartar información importante de la clase mayoritaria.

Sobremuestreo (Oversampling):

Aumenta el tamaño de la clase minoritaria generando nuevas muestras o duplicando las existentes. Aprovecha toda la información de la clase mayoritaria, pero podría generar un sobreajuste.

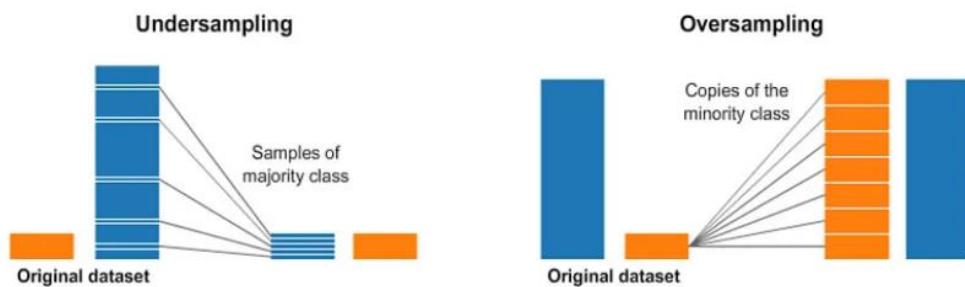


Figura 1.13 Ejemplificación del sub muestreo y sobre muestreo.

1.5.6.6 Escalamiento de datos

Dado que las características pueden tener diferentes rangos de valores, las que tengan un rango mayor pueden dominar el cálculo de las distancias o los gradientes, haciendo que el modelo se sesgue. El escalamiento ayuda a estandarizar o normalizar los datos para que todas las características tengan un impacto similar en el modelo.

La elección del método de escalamiento depende del algoritmo de machine learning que se use y de las características de los datos. Por ejemplo, los modelos basados en gradiente como redes neuronales o regresión logística suelen beneficiarse de la estandarización. Por otro lado, redes neuronales profundas a menudo prefieren la normalización (Min-Max) en un rango específico.

Por último, modelos basados en distancia se benefician del escalamiento por norma.

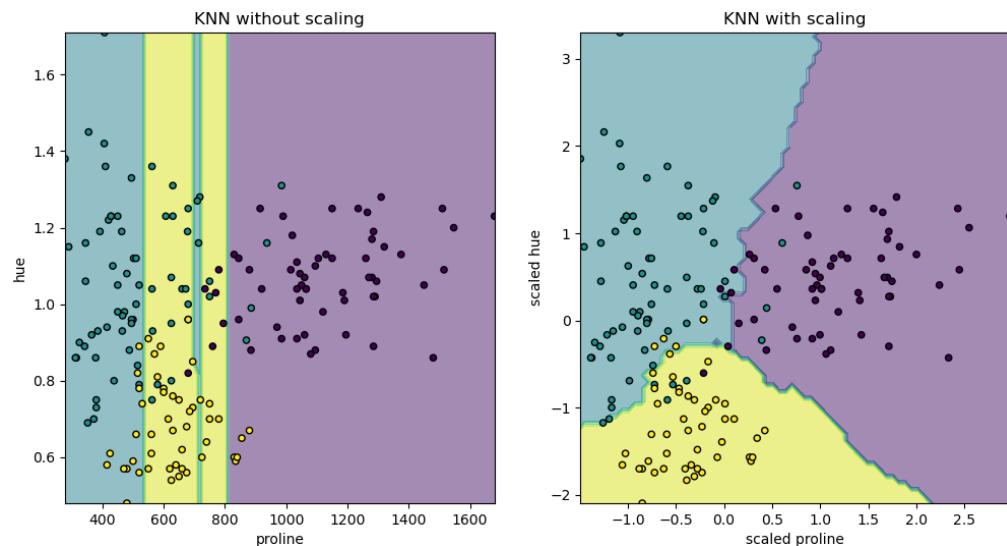


Figura 1.14 Mejora en el tratamiento de datos cuando se usa escalamiento.

1.5.6.7 Matriz de confusión

La matriz de confusión es una herramienta fundamental en la evaluación de modelos de clasificación en machine learning, ya que permite visualizar el desempeño del modelo al clasificar instancias en diferentes clases.

Esta matriz muestra la relación entre las predicciones del modelo y las clases reales de los datos, lo cual facilita identificar los errores específicos y analizar cómo se comporta el modelo en cada clase.

Además, ayuda a diagnosticar errores específicos. Por ejemplo, si el modelo tiene muchos falsos negativos, puede que sea necesario ajustar la sensibilidad del modelo, mientras que, si tiene muchos falsos positivos, se podría priorizar la precisión.

		Predicted values	
		Positive	Negative
Actual Values	Positive	True Positive [TP]	False Negative [FN] (Type II Error)
	Negative	False Positive [FP] (Type I Error)	True Negative [TN]

Figura 1.15 Matriz de confusión.

1.5.6.8 Cross Validation

La validación cruzada es una técnica utilizada para evaluar el rendimiento de un modelo de machine learning de manera más robusta y evitar el sobreajuste (overfitting).

En lugar de entrenar y probar el modelo en una sola partición de los datos, la validación cruzada divide los datos en varias particiones para evaluar el modelo en diferentes subconjuntos, proporcionando una evaluación más confiable de su rendimiento.

Esta técnica es esencial para asegurar que el modelo funcione bien en nuevos datos y no solo en el conjunto de datos de entrenamiento. Además aporta una forma más confiable de evaluar el rendimiento, reduciendo el riesgo de que el modelo presente un alto error cuando se despliegue en producción.



Figura 1.16 Alternación característica en la data propia del Cross Validation.

1.5.6.9 Análisis componentes principales

El Análisis de Componentes Principales (PCA, por sus siglas en inglés) es una técnica de reducción de dimensionalidad ampliamente utilizada en machine learning y análisis de datos. Su objetivo principal es transformar un conjunto de variables posiblemente correlacionadas en un conjunto más pequeño de variables no correlacionadas llamadas "componentes principales", que explican la mayor variabilidad posible de los datos originales.

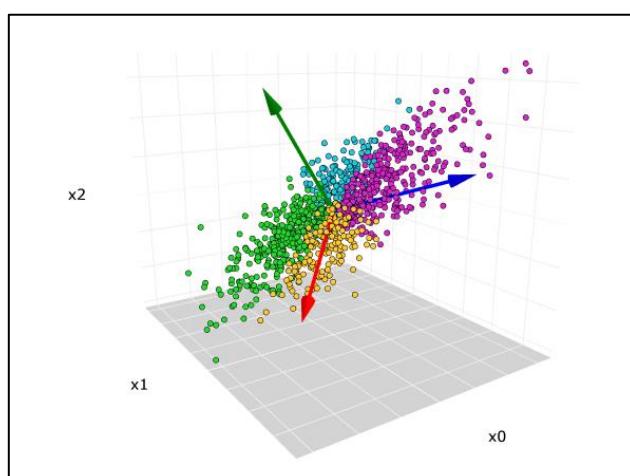


Figura 1.17 Descomposición en dimensiones luego de aplicar PCA.

CAPÍTULO 2

2. METODOLOGÍA

Se muestra la metodología para hacer la evaluación de la dinámica de la red y evaluar casos de probable riesgo en base a su estabilidad oscilatoria.

Se tomará la red peruana y sobre ella se realizarán reducciones para disminuir la exigencia computacional. Seguido a ello se realizará simulación de contingencias off-line mediante script de PowerFactory, generando data del sistema, tales como tensión, corriente, potencia y amortiguamiento del sistema.

Usando esta data se usa algunos clasificadores inteligentes, de los cuales se escoge el de mejor rendimiento y se tiene un modelo entrenado y probado. La idea es mostrar que la evaluación rápida de la estabilidad oscilatoria del sistema peruano mediante la clasificación de data es factible y que a futuro se aplique en tiempo real para el sistema peruano usando la data del sistema Scada – WAMS.

Se muestra una secuencia de la metodología en el siguiente grafico:

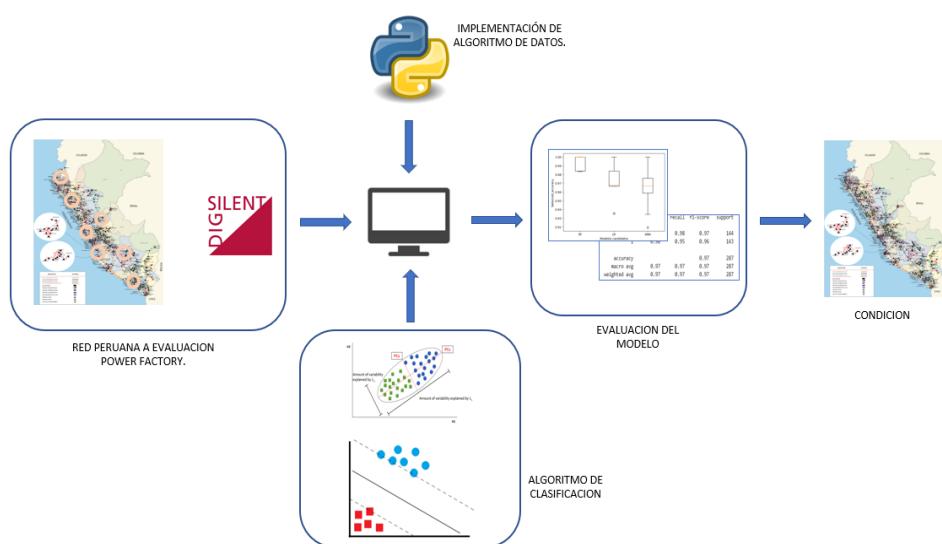


Figura 2.1 Esboce gráfico de la metodología planteada.

También queda representado mediante un diagrama de flujo:

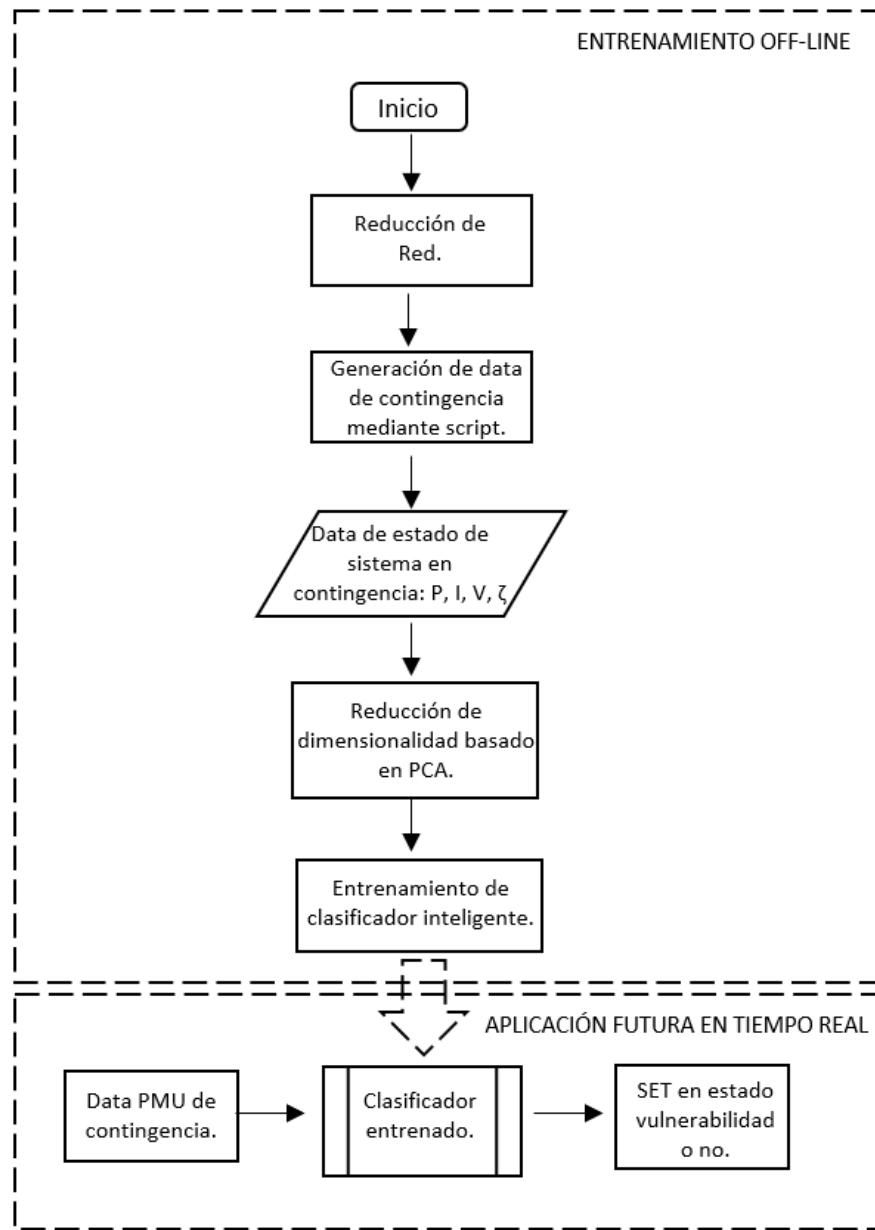


Figura 2.2 Metodología de Identificación de vulnerabilidad de la red peruana.

En las siguientes líneas se explica el proceso interno de generación de data para entrenamiento dentro del diagrama de flujo presentado:

Mediante un script se toma elementos de red y se generan contingencias N-1 y N-2 (líneas de transmisión). Esto considerando al hecho práctico de que en la red peruana al

tener un elemento fuera de línea, muchas veces el operador da permiso a sacar otra por mantenimiento, aumentando así la exigencia del sistema.

Seguidamente se calcula si la variación de amortiguamiento de la red es mayor a un valor determinado, además de la cantidad de veces que se supera dicho criterio. Para cada estado que cumpla dicho criterio, se toma data de las variables de tensión, corriente, potencia y amortiguamiento que los conlleva, los cuales son tomados y guardados en un archivo csv. Esto se repite para todas las contingencias en base a los elementos a salir seleccionados.

Lo descrito se presenta en el siguiente diagrama de flujo:

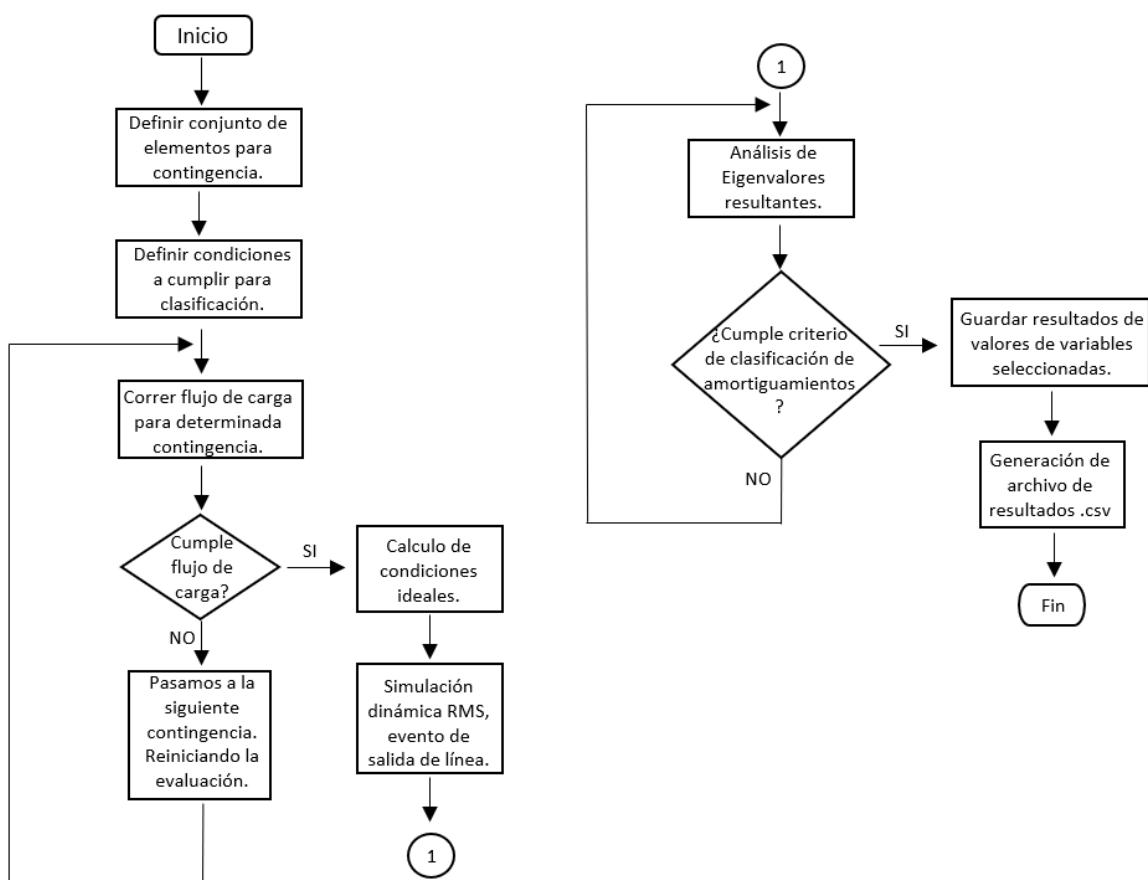


Figura 2.3 Diagrama de flujo para el scripting de PowerFactory.

En los siguientes incisos se detalla de forma extendida la metodología utilizada.

2.1 Reducción de red y generación de data del sistema peruano

Para la reducción de la red peruana se hace uso de la herramienta de PowerFactory llamada Boundary – Network Reduction, el cual se muestra en la figura 2.4. Esta se encuentra dentro de las herramientas “Aditional Function”.

El análisis se centra en el método equivalente de Ward y se centra hacer reducciones para las fronteras que se crean de acuerdo a la frontera A_Frontera1, el cual luego de múltiples reducciones nos resulta en una red aún más reducida.

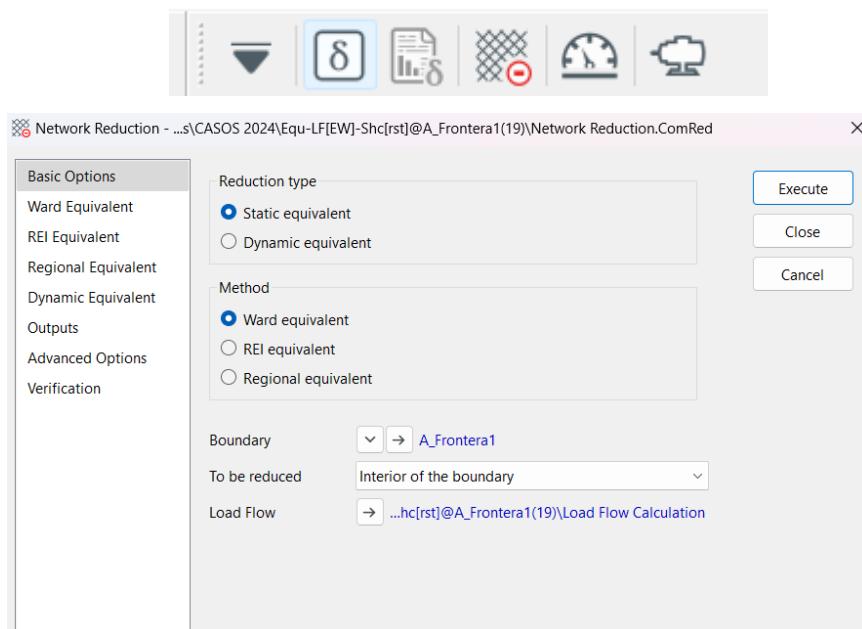


Figura 2.4 Herramienta para reducción de redes – Network Reduction

Se hizo una iteración de 20 reducciones, y mediante el flujo de carga y condiciones iniciales se valida la convergencia de la red.

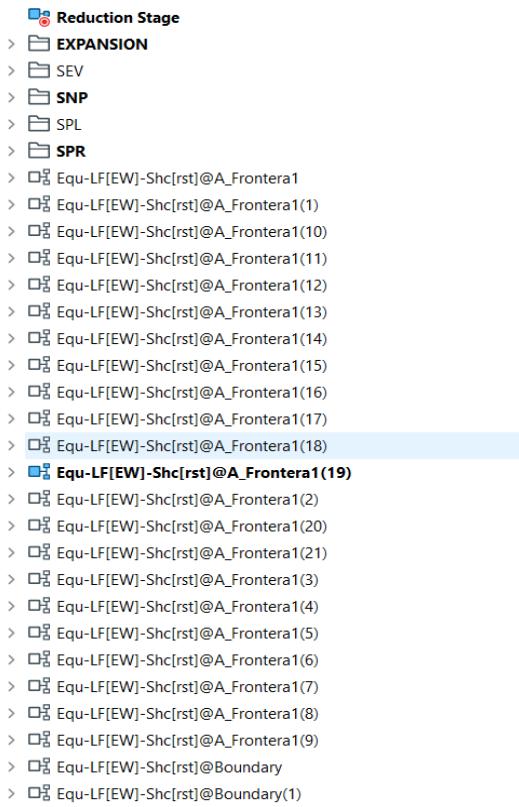


Figura 2.5. Boundaries o fronteras realizadas para la reducción de red.

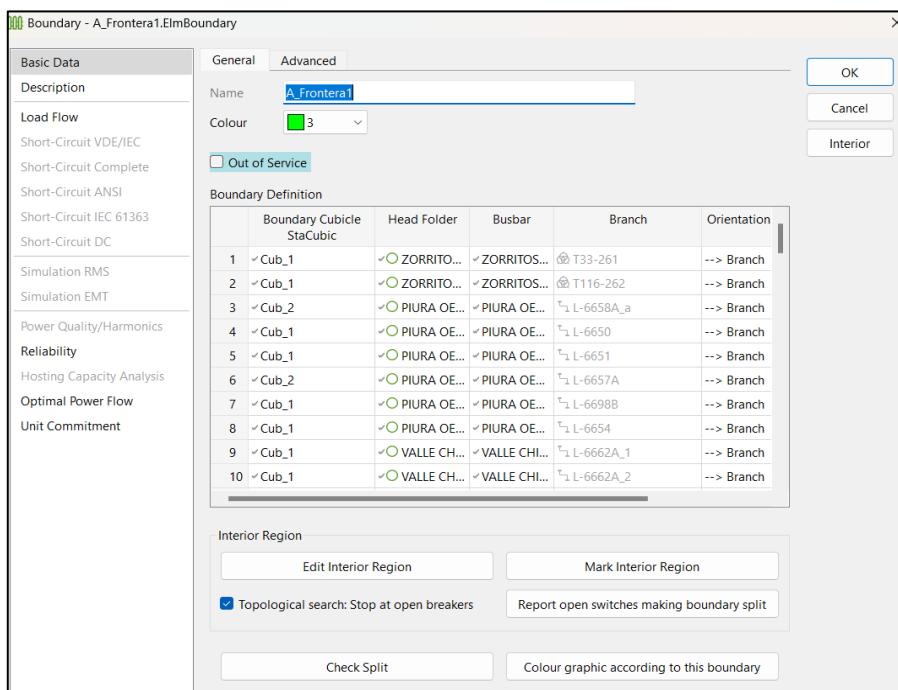


Figura 2.6 Descripción de fronteras para reducciones.

Una vez realizados, se ejecuta la reducción de red en base y obtenemos la red reducida.

Por otro lado, se escogen las barras o líneas más importantes del sistema:

	Name	Order	Type	Object modified
▶	L-2249	0.	220_Talara-Zorritos_L2249_1...	30/11/2022 16:55:06
▶	ZORRITOS 220A	0.		20/07/2023 14:17:59
▶	L-2295	1.	220_Talara-PiuraOeste_L224...	30/11/2022 16:55:06
▶	TALARA 220A	1.		20/07/2023 13:13:42
▶	L-2241	2.		30/11/2022 16:55:06
▶	PARIÑAS 220A	2.		18/08/2023 16:02:26
▶	L-5010	3.	500_Trujillo-Niña	30/11/2022 16:55:06
▶	LA NIÑA 500A	3.		18/08/2023 16:02:26
▶	L-2239_A	4.		30/11/2022 16:55:06
▶	TRUJILLO 500A	4.		18/08/2023 16:02:26
▶	L-2236	5.		30/11/2022 16:55:06
▶	TRUJILLO NORTE 220A	5.		20/07/2023 13:15:15
▶	L-2190	6.	Lne_Carhuacero-Cajamarc...	30/11/2022 16:55:06
▶	CHIMBOTE 500A	6.		18/08/2023 16:02:26
▶	L-5008	7.	500_Chimb-Truj(1c) L-5008	30/11/2022 16:55:06
▶	CHIMBOTE 2 220A	7.		20/07/2023 13:15:41
▶	L-5006	8.	500_Carab-Chimb(1c) L-5006	30/11/2022 16:55:06
▶	CARABAYLLO 500A	8.		18/08/2023 16:02:26
▶	L-2216	9.	220_Chimbote-Paramonga...	30/11/2022 16:55:06
▶	CARABAYLLO 220A	9.		20/07/2023 13:26:10
▶	L-5003_B	10.		17/08/2023 08:48:44

Figura 2.7 Líneas y barajes seleccionados.

Seguido a eso se genera la data mediante un scripting en PowerFactory:

```

DPL Command - Scripts(Comp_Eigen.ComDpl)
Basic Options
Results
Script
Description
Version
Selecting a remote script destroys all current settings
Remote script
Code
Execute
Close
Cancel
Save
Check
Encrypt
Contents
70 while (oLine);
71 {
72     oLine=oLineo.First();
73     for (f=1;f<oLineasf+=1)
74     {
75         iFOR;
76         oLine=oLineo.srv;
77         printf("F/S SEC: %s",oLine.oLoc_name);!Sacamos fuera de servicio linea
78         oLineo.outserv=1;
79         oLineo.outserv=1;
80         printf("F/S SEC: %s",oLine.oLoc_name);!Sacamos de servicio linea j
81         oLineo.outserv=1;
82         printf("F/S SEC: %s",oLine.oLoc_name);!Sacamos de servicio linea j
83     }**FIN 2. FERO SE DEJA ABIERTO EL BUCLE DE LINEAS j
84
85 !INICIO 3: ***CONTENEMOS VALORES DE AMORTIGUAMIENTO EN ESTA CONTINGENCIA
86
87 conting.Init(h);
88 contingOrd.Init(h);
89
90 EchoOff();
91
92 error_ldf=ldf.Execute();
93
94 if (error_ldf!=0){obtencion de contingencia y comparacion (y su guardado
95
96 !vIREmos con los valores de la contingencia
97
98
99 Modal_exe=Modal_Execute();
100 str="Modal analysis executed";
101 str+="\n";
102 LoadResData(Eigenvalores1);
103 Neig = ResVal(Eigenvalores1,0); !Returns the number of values stored
104 str+="Neig";

```

Figura 2.8 Scripting – PowerFactory.

La estructura de la data generada para entrenamiento que arroja el script es de la siguiente forma:

Tabla 1.1 Estructura de la data generada para entrenamiento.

	ENTRADAS							SALIDAS
	X							Y
	Elemento Line 1		Elemento Line 2		...N	Elemento Bus 1		...N
	Potencia	Corriente	Potencia	Corriente	--	Tensión	Angulo	--
Contingencia 1
Contingencia 2
Contingencia 3
Contingencia 4
...
...
...
Contingencia i-2
Contingencia i-1
Contingencia i

Donde se aprecia las entradas y salidas de la matriz de entrenamiento que ingresaran al modelo de entrenamiento Python.

2.2 Proceso de entrenamiento de data

La base de datos obtenidas de PowerFactory de DlgSILENT entrega escenarios dependiendo del cumplimiento de los criterios de evaluación de seguridad programados en el script.

Durante el desarrollo del presente proyecto se identificó la necesidad de realizar una clasificación de estos escenarios, ya que hay escenarios en los cuales el sistema se encuentra en la capacidad de mantener condiciones que no superan límites del criterio

de seguridad y otros en que es necesario considerarlos como probables. Por lo que se usarán algoritmos del tipo clasificación.

El clasificador se programa en Python y permitirá entrenar y clasificar los escenarios identificando si existe o no la necesidad de alertar sobre la condición del sistema. Cabe indicar que previamente la data debe dársele algún tipo de tratamiento de dimensionalidad, escalamiento y limpieza de datos.

2.2.1 Escalamiento, reducción de dimensionalidad y cross validation

Con el escalamiento estandarizamos o normalizamos los datos para que todas las características tengan un impacto similar en el modelo. Se usará una técnica llamada StandardScaler.

Por otro lado, la data recibida de sistemas tan grandes como el peruano tiene muchísimas variables disponibles a escoger para análisis. Se tomará una cantidad de variables que se pueden obtener, es decir, estar disponibles en la práctica. Tales como tensiones, corrientes y potencias. Pero esta cantidad afectara al cálculo debido a la cantidad de procesamiento requerido, por lo cual aplicaremos la técnica de componentes principales para disminuir dimensionalidad.

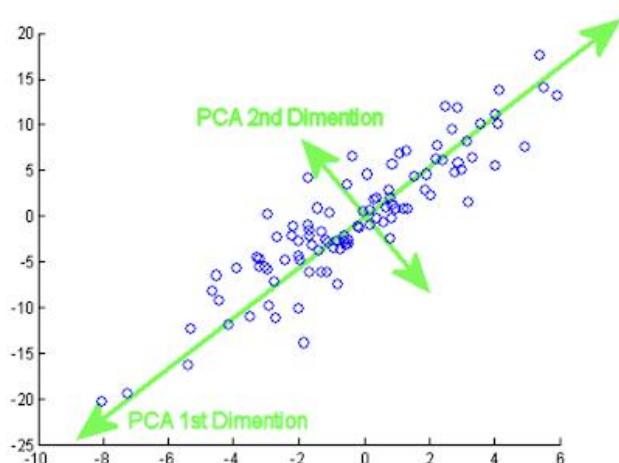


Figura 2.9 Diagrama de flujo para el scripting de PowerFactory.

La cual como se puede apreciar en la tabla 1.2, la gran cantidad de data del sistema peruano puede reflejarse (transformarse) en determinada menor cantidad debido a la dimensionalidad. Normalmente buscaremos reducir en una cantidad que contemple la mayor parte de la característica. Aplicando un análisis tipo codo encontraremos la reducción adecuada.

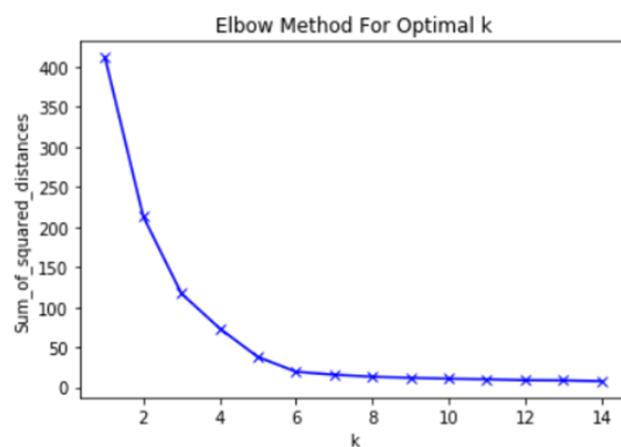


Figura 2.10 Método del codo.

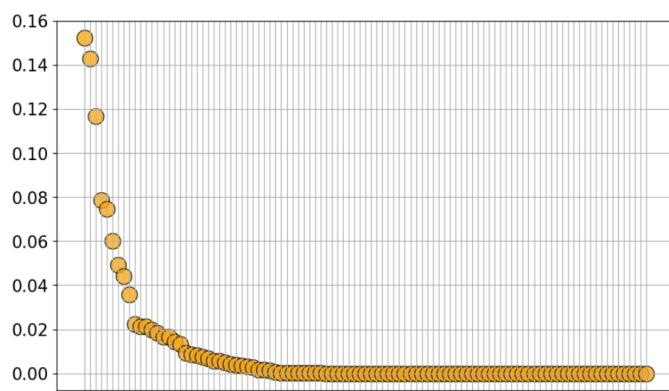


Figura 2.11 Cantidad de componentes principales a elegir mediante método de codo.

Adicionalmente se considera la técnica del Cross Validation para aprovechar el mejor entrenamiento cruzado posible:

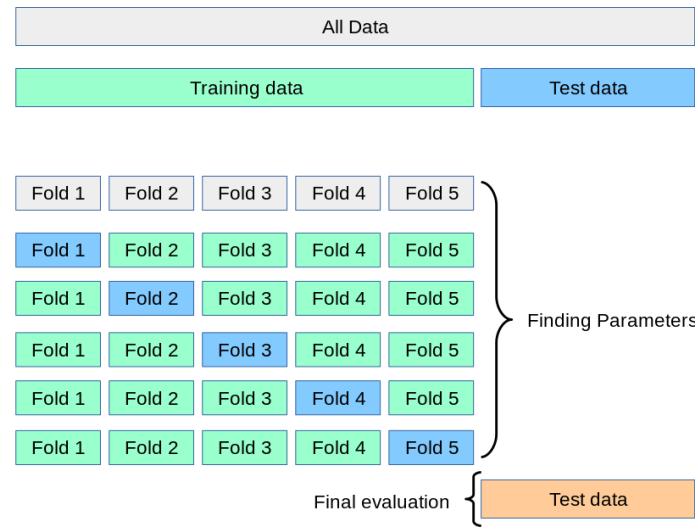


Figura 2.12 Técnica del cross validation.

```
# Entreno con validación cruzada
for scoring in error_metrics:
    print('Métrica de evaluación: ', scoring)
    for name, model in models:
        print('Modelo ', name)
        cross_validation = StratifiedKFold(n_splits=num_folds, random_state=0, shuffle=True)
        cv_results = cross_val_score(model, X_train_scaled, y_train, cv=cross_validation, scoring=scoring)
        #print(cv_results)
        results.append(cv_results)
    if ite == 0:
        names.append(name)
    resume = (name, cv_results.mean(), cv_results.std())
    msg.append(resume)
print(msg)
```

Figura 2.13 Técnica del cross validation aplicado a los datos generados.

2.2.2 División de los datos

Para fines del entrenamiento y la validación de los datos obtenidos, es que la data se separa en datos de entrenamiento, validación y prueba.

Se sigue la buena práctica de escoger un 80% de datos para entrenamiento y 20% para prueba. Lo mismo para dividir los datos de entrenamiento y validación.

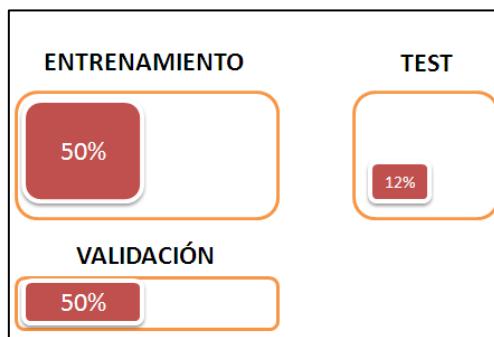


Figura 2.14 Ejemplo de división de cierta data.

2.2.3 Predicción mediante modelos y comparación

Para este punto se entrenarán y evaluarán diferentes tipos de modelos. Para validar si un modelo es bueno, una buena práctica es compararlo respecto de otros, evaluando el de mejor resultado, menor varianza y cantidad de outliers.

Muchas veces un determinado modelo, debido a la concepción de su algoritmo nos genera resultados más exactos respecto de otros, debido a que la data se acomoda a la forma de ciertos algoritmos y su modo lógico-matemático de procesarlos.

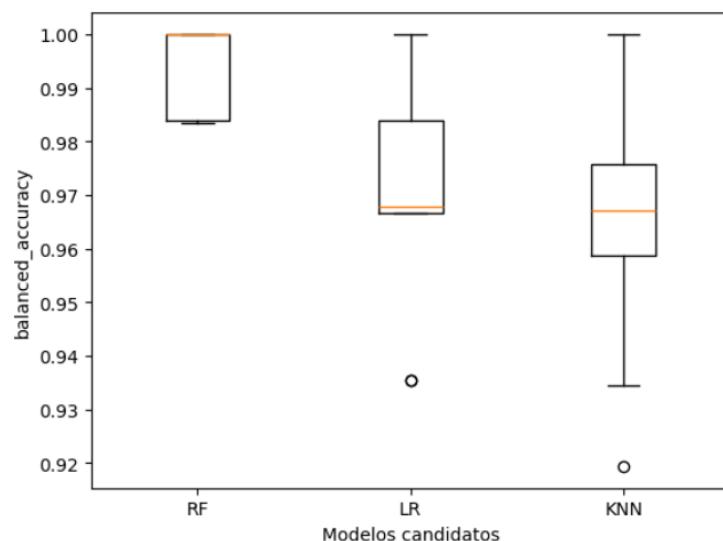


Figura 2.15 Comparación de modelos de entrenamiento.

2.3 Viabilidad técnica

Esta tecnología requiere elementos necesarios para una implementación práctica en campo. Como se observa en la siguiente figura, para el funcionamiento de esta se requieren IEDs, enlaces de comunicaciones y un preparado centro de control (servidores, pdc, algoritmos entrenados, personal capacitado, etc).

Se presenta una descripción final de la tecnología que acompaña para un mejor entendimiento de lo desarrollado.

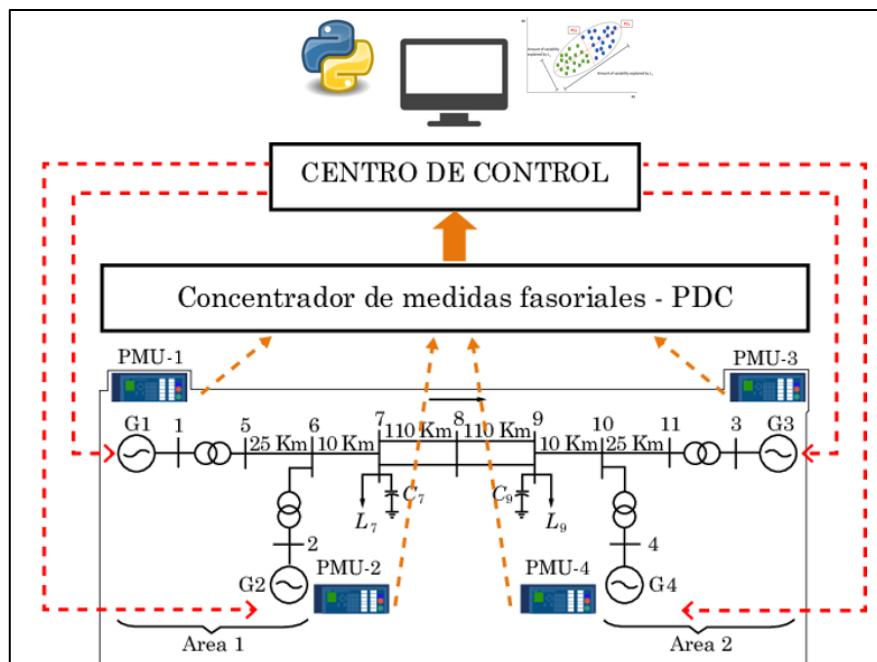


Figura 2.16 Elementos necesarios para implementación de algoritmos IA aplicados.

CAPÍTULO 3

3. RESULTADOS Y ANÁLISIS

En este capítulo se presentan los resultados obtenidos de la aplicación integral de la metodología. Así, se presentan primero la data generada en base al algoritmo desarrollado en PowerFactory primeramente para después pasar a la aplicación del aprendizaje máquina. Se muestran resultados de los clasificadores para la red, así como la comparación de éstos con técnicas de minería de datos como la regresión logística, árboles de decisión y K vecinos más cercanos.

3.1 Preparación de la red

3.1.1 Reducción del Sistema peruano

Se tomo la red representativa del SEIN para el año 2024:

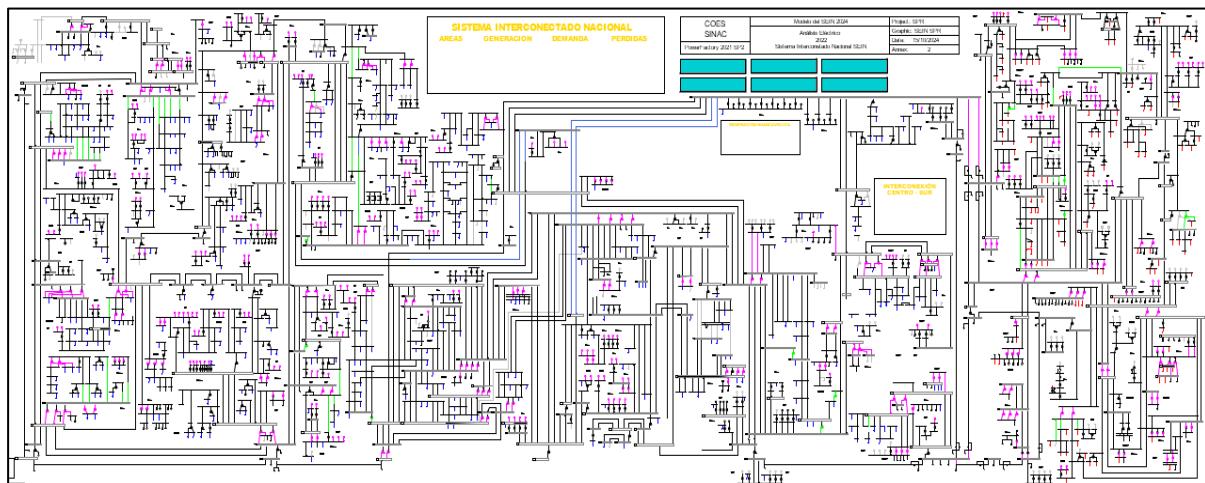


Figura 3.1 Red representativa del sistema eléctrico peruano.

Como se aprecia, la red del SEIN es bastante extensa y mallada entre sí, lo cual nos complica el requerimiento de esfuerzo computacional y manejar una gran cantidad de variables.

Se identifica las áreas que se puedan reducir y no son parte de las troncales más importantes del país. Mediante la herramienta Network Reduction del PowerFactory se reducen las áreas, identificadas de la forma siguiente:

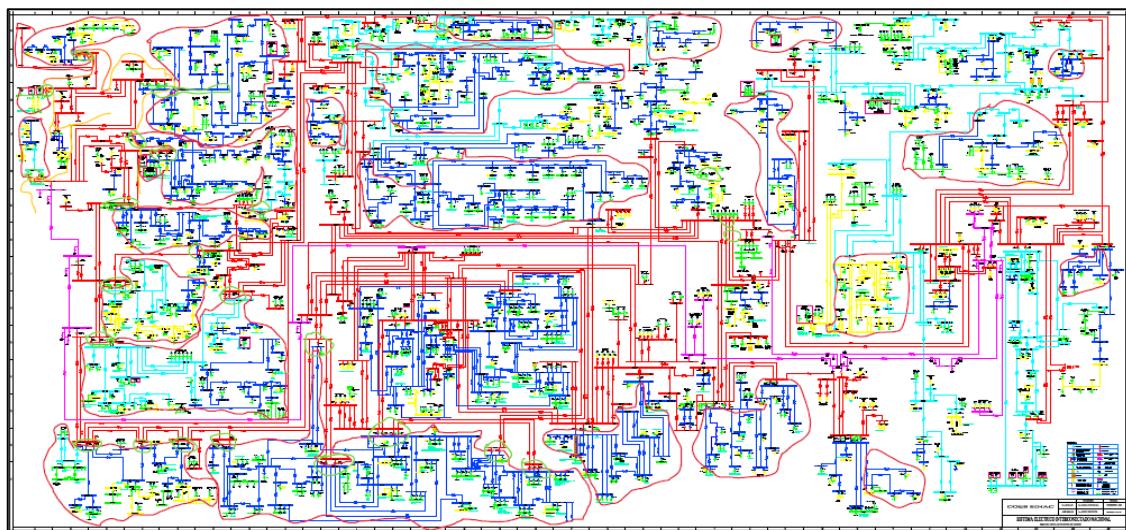


Figura 3.2 Áreas a reducir.

Lo cual nos resulta en la red siguiente:

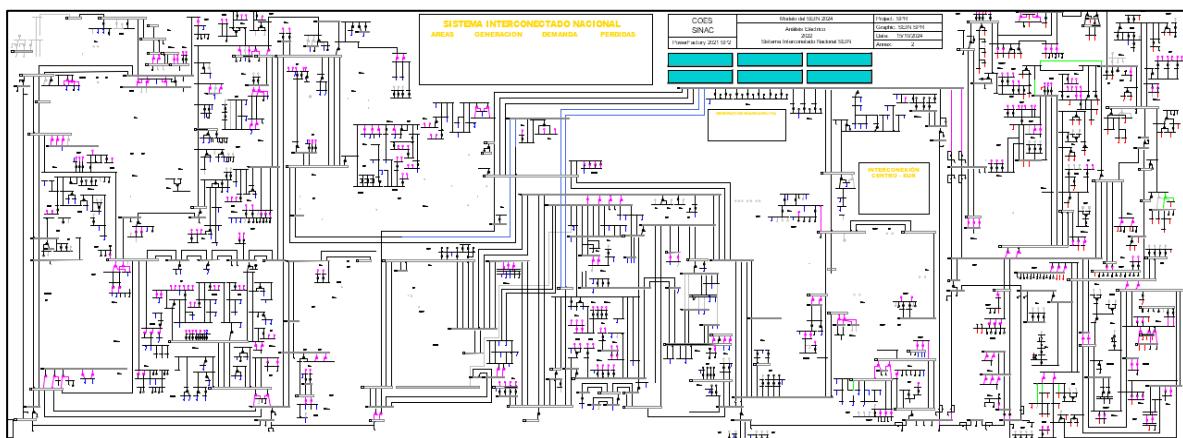


Figura 3.3 Red reducida.

3.1.2 Elección de elementos del sistema eléctrico

Para la selección de las líneas y barras más representativas del SEIN se hizo la elección de acuerdo al Estudio de estabilidad del SEIN 2022 – 2025 del SEIN elaborado por COES [13].

A continuación, se muestra en la siguiente figura:

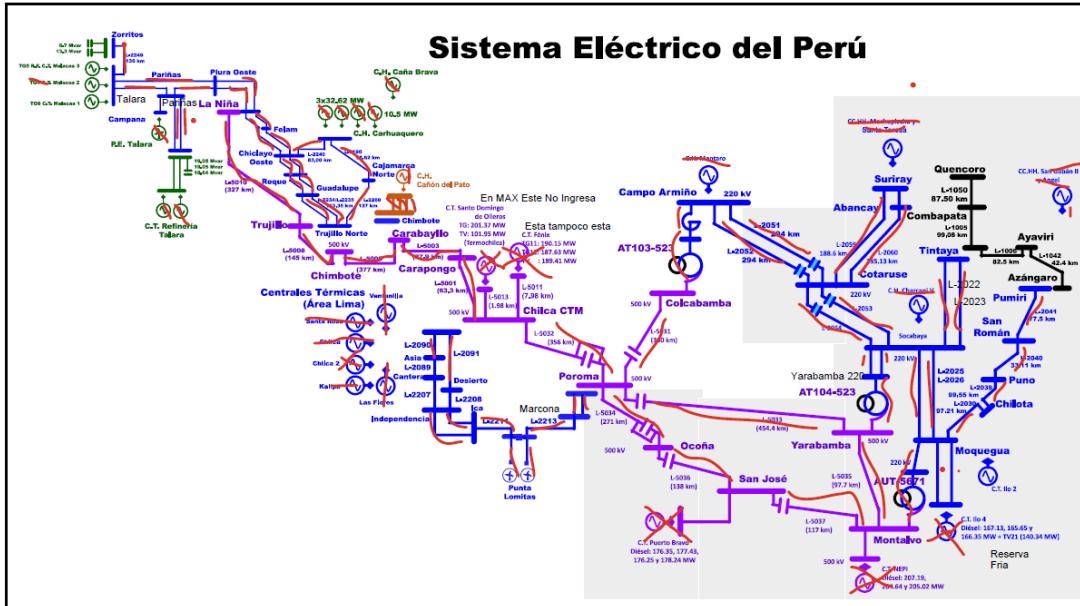


Figura 3.4 Trazos de los elementos de red representativos del sistema eléctrico peruano.

Para el análisis se ha considerado las siguientes líneas de transmisión que son consideradas importantes y de mayor relevancia de acuerdo a [13]:

Tabla 2.2 Lineas de transmisión en análisis.

Order	Name	Type
1	L-2249	220_Talara-Zorritos_L2249_152MVA
2	L-2295	220_Talara-PiuraOeste_L2248-New
3	L-2241	220_Piura Oeste - La Niña
4	L-5010	500_Trujillo-Niña
5	L-2239_A	La Niña 220 - Derivacion
6	L-2236	220_Reque - Guadalupe
7	L-2190	Lne_Carhuacuero-Cajamarca220
8	L-5008	500_Chimb-Truj(1c) L-5008
9	L-5006	500_Carab-Chimb(1c) L-5006
10	L-2216	220_Chimbote-ParamongaNueva_L2215
11	L-5003_B	500_La Niña - Carabayllo
12	L-5001_A	500_Chilca - Carabayllo
13	L-2209	220_Independencia-Ica segunda terna
14	L-2090	220_Chilca-Cantera_L2090
15	L-2292	220_Marcona Nueva-Marcona(1)
16	L-5034	500_Poroma - Ocoña

17	L-5031	Lne Colcabamba-Poroma 500 kV
18	L-5033	Lne Colcabamba-Poroma 500 kV
19	L-5035	Lne Yaramamba-Montalvo 500 kV
20	L-5037	500_San Jose - Montalvo
21	L-2052	Lne Campo Armiño
22	L-2060	220_L-2050-L-2059-L-2060
23	L-2065	Lne Yarabamba-Socabaya 220 kV
24	L-2022	220_Tintaya Nueva - Socabaya
25	L-2025	220_Socabaya-Moquegua_L2026
26	L-2040	Puno - San Román
27	L-2036D	220_Derivacion - Montalvo
28	L-2278	220_Vizcarra-ParamongaNueva_L2253(1)
29	L-2252	220_TingoMaria-Vizcarra_L2252(1)
30	L-2234	220_Guadalupe-TrujilloNorte_L2234
31	L-2057	220_Montalvo 2-Montalvo
32	L-2132_a	220_Chiribamba - Independencia
33	L-2297	220_Reque - Chiclayo Oeste
34	L-2211	220_Ica-Marcona_L2211-New
35	L-2231	220_Huancavelica - Independencia

Asimismo, para los siguientes barajes:

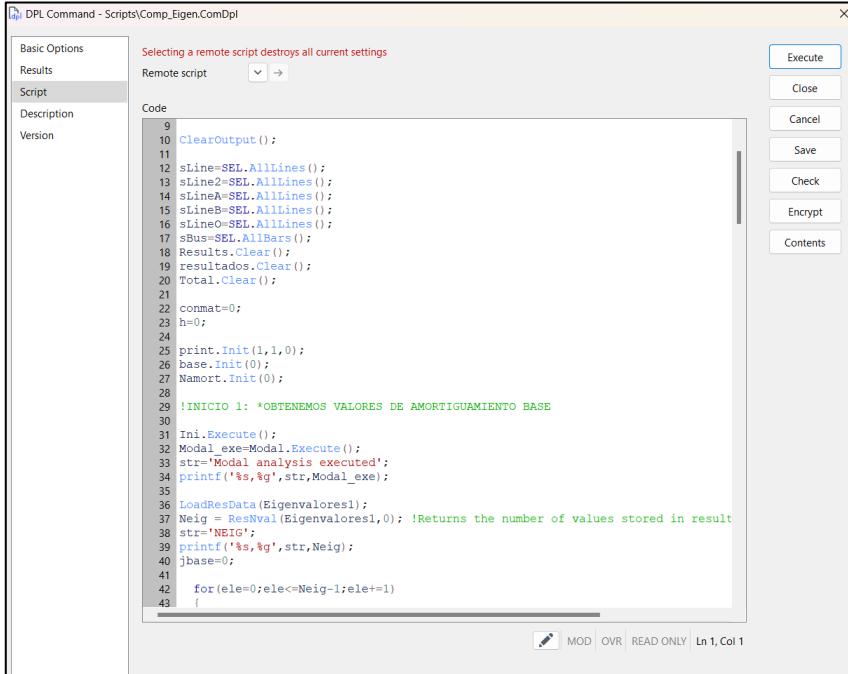
Tabla 3.3 Barras de transmisión en análisis

Order	Name
0	ZORRITOS 220A
1	TALARA 220A
2	PARIÑAS 220A
3	LA NIÑA 500A
4	TRUJILLO 500A
5	TRUJILLO NORTE 220A
6	CHIMBOTE 500A
7	CHIMBOTE 2 220A
8	CARABAYLLO 500A
9	CARABAYLLO 220A
10	CHILCA CTM 500A
11	CHILCA CTM 220A
12	POROMA 500A
13	POROMA 220A
14	COLCABAMBA 500A
15	COLCABAMBA 220A
16	OCOÑA 500A

17	OCOÑA 60kV
18	SAN JOSE 220A
19	SAN JOSE 500A
20	YARABAMBA 220A
21	YARABAMBA 500A
22	Montalvo 500_L-5037
23	CAMPO ARMIÑO 220A
24	COTARUSE1 220A
25	SOCABAYA 220A
26	MOQUEGUA 220A
27	MARCONA 220A
28	INDEPENDENCIA 220A

3.2 Generación de datos

Luego mediante el software PowerFactory se realiza la generación de data para entrenamiento de modelos. Se usará el scripting de PowerFactory.



The screenshot shows a software interface titled "DPL Command - Scripts\Comp_Eigen.ComDpl". The left sidebar has tabs for "Basic Options", "Results", "Script" (which is selected), "Description", and "Version". The main area is titled "Selecting a remote script destroys all current settings" and contains a dropdown menu "Remote script" with two options. Below this is a "Code" section containing a script in DPL language. The script initializes variables, performs modal analysis, and loads results. On the right side, there are buttons for "Execute", "Close", "Cancel", "Save", "Check", "Encrypt", and "Contents". At the bottom, there are status indicators: "MOD", "OVR", "READ ONLY", and "Ln 1, Col 1".

```

9
10 ClearOutput();
11
12 sLine=SEL.AllLines();
13 sLine2=SEL.AllLines();
14 sLineA=SEL.AllLines();
15 sLineB=SEL.AllLines();
16 sLineO=SEL.AllLines();
17 sBus=SEL.AllBuses();
18 Results.Clear();
19 resultados.Clear();
20 Total.Clear();
21
22 connmat=0;
23 h=0;
24
25 print.Init(1,1,0);
26 base.Init(0);
27 Namort.Init(0);
28
29 !INICIO 1: *OBTEMOS VALORES DE AMORTIGUAMIENTO BASE
30
31 Ini.Execute();
32 Modal_exe=Modal.Execute();
33 str='Modal analysis executed';
34 printf('%s,%g',str,Modal_exe);
35
36 LoadResData(Eigenvalores1);
37 Neig = ResNVal(Eigenvalores1,0); !Returns the number of values stored in result
38 str='NEIG';
39 printf('%s,%g',str,Neig);
40 jbase=0;
41
42 for(ele=0;ele<=Neig-1;ele+=1)
43 {

```

Figura 3.5 Script desarrollado para el análisis.

Se excita la dinámica del sistema mediante eventos de salida de servicio de líneas. Mediante la programación se generan escenarios con contingencias N-1 y N-2, las cuales son frecuentes debido a trabajos de mantenimientos programados a lo largo del sistema.

Esto permite exponer que los modos de oscilación del sistema para cada escenario y evaluar los cambios de estos valores, y los valores de parámetros que están asociados al patrón de estos valores.

Se genera una matriz de datos de 956 x 102 los cuales se usarán para el entrenamiento. Los datos se generaron bajo la premisa de que si el amortiguamiento de la red varía en más de 10% y tenemos por lo menos 40 modos que superan esa diferencia, el script captura los valores del parámetro del sistema.

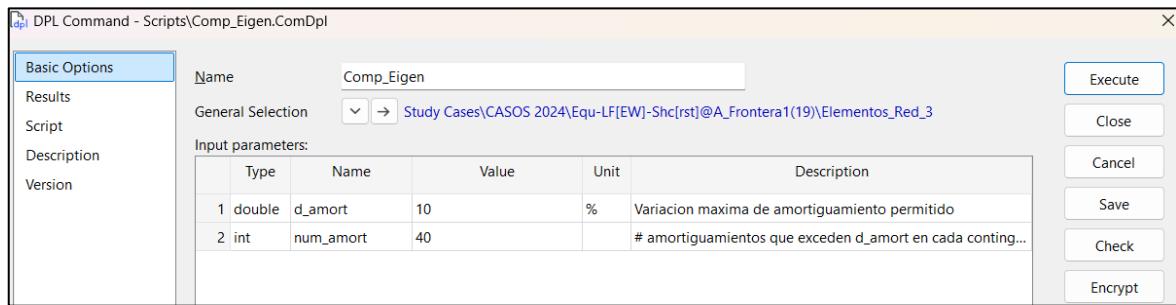


Figura 3.6 Criterios relacionados a eigenvalores para captura de parámetros.

Y esto genera la siguiente matriz descrita de forma comprimida:

Tabla 4.4 Data Generada.

Results	L-2022	L-2022	L-2040	L-2040	L-2060	...	L-2052	L-2052	L-2252	L-2252	Comp_Eigen	Clasificación
Index	Current, Magnitude/ Power/Term	Active Current, Magnitude/ Power/Term	Current, Magnitude/ Power/Term	Active Current, Magnitude/ Power/Term	Current, Magnitude/ Power/Term	...	Current, Magnitude/ Power/Term	Active Current, Magnitude/ Power/Term	Active Power/Terminal i in MW	Amortiguam iento		
0	0.186363	-68.034225	0.025704	-6.703865	0.108079	...	0.616442	249.639803	0.123459	-46.424479	7.638103	1
1	0.18634	-68.035196	0.025727	-6.677561	0.108241	...	0.615425	249.292956	0.121132	-45.419867	8.059595	1
2	0.186545	-68.084164	0.025018	-6.621664	0.10851	...	0.610344	247.013942	0.113472	-42.074772	7.790017	1
3	0.191821	-69.391841	0.043886	-14.929118	0.083565	...	0.785933	314.692916	0.12506	-47.078453	8.226663	1
4	0.179596	-67.870715	0.035039	-12.678257	0.072376	...	0.872687	345.639413	0.124441	-46.774669	7.677451	1
5	0.150135	-66.022997	0.032998	-5.550135	0.118927	...	0.815357	334.78425	0.124996	-47.107993	8.235123	1
6	0.18635	-67.985012	0.026172	-6.929302	0.106768	...	0.627723	253.979001	0.121342	-45.512409	7.638162	1
7	0.186507	-68.039911	0.025465	-6.807845	0.107429	...	0.619874	250.702261	0.109176	-40.213269	7.511835	1
8	0.190732	-69.302844	0.044767	-14.95782	0.084869	...	0.793897	317.330064	0.123619	-46.447443	8.197496	1
9	0.178956	-67.796888	0.034436	-12.623354	0.071276	...	0.88167	346.774263	0.123057	-46.149871	8.276019	1
10	0.147581	-66.067087	0.026712	-5.664602	0.116078	...	0.829884	337.071987	0.123624	-46.487768	8.212925	1
11	0.18631	-67.987332	0.026191	-6.889219	0.107008	...	0.626118	253.417	0.118767	-44.389104	7.999234	1
12	0.188176	-70.530618	0.04568	-16.709431	0.076432	...	0.810863	320.476321	0.123624	-46.424515	7.855575	1
13	0.146406	-66.157676	0.022315	-5.753228	0.116214	...	0.835599	338.377841	0.123538	-46.444048	7.46278	1
14	0.186363	-68.034225	0.025704	-6.703865	0.108079	...	0.616442	249.639803	0.123459	-46.424479	7.638103	1
15	0.186341	-67.985012	0.026172	-6.929302	0.106768	...	0.627723	253.979001	0.121342	-45.512409	7.638162	1
16	0.186338	-67.984296	0.026183	-6.929582	0.106767	...	0.627778	254.005899	0.121333	-45.506956	7.638085	1
17	0.186335	-67.967686	0.026289	-6.972416	0.105973	...	0.631175	254.894255	0.120282	-45.278194	7.637283	1
18	0.186369	-67.987817	0.026111	-6.941548	0.106696	...	0.627986	254.036964	0.121817	-45.721079	7.908495	1
19	0.186356	-68.032779	0.025707	-6.687037	0.107928	...	0.616271	249.403036	0.118305	-44.200472	7.914927	1
20	0.186437	-68.04705	0.025466	-6.687683	0.107853	...	0.615473	248.925642	0.12079	-45.257324	7.63819	1
21	0.186358	-67.988976	0.026114	-6.926046	0.106791	...	0.627345	253.811586	0.121334	-45.509196	7.638085	1
22	0.186344	-67.991514	0.026106	-6.8962	0.106914	...	0.626174	253.349401	0.121404	-45.536925	7.638083	1
23	0.18634	-67.977812	0.026239	-6.96474	0.10657	...	0.629456	254.648517	0.121324	-45.502328	7.638085	1
24	0.186703	-70.560644	0.045249	-16.751873	0.076864	...	0.811024	320.660506	0.121744	-45.625699	7.578212	1
25	0.202568	-73.452755	0.051495	-18.591003	0.103657	...	0.621449	250.913984	0.121296	-45.481251	7.63808	1
26	0.187518	-68.421508	0.019148	-4.149116	0.102537	...	0	0	0.12095	-45.323727	7.638033	1
...
946	0	0	0.041798	10.259743	0.09987	...	0.635385	257.176352	0.123149	-46.276961	8.912723	0
947	0.193865	-70.869071	0.039425	-13.040014	0.106655	...	0.622649	252.116334	0.123124	-46.266019	16.069854	0
948	0.186131	-66.923877	0	0	0.107072	...	0.631635	255.706002	0.123152	-46.278894	16.070414	0
949	0.186436	-68.003964	0.026517	-7.118854	0.106447	...	0.632377	256.004418	0.123153	-46.279546	16.070341	0
950	0.186206	-67.943379	0.026722	-7.012312	0.106319	...	0.633359	256.360519	0.126603	-47.770386	12.954964	0
951	0.186211	-67.950892	0.026649	-6.981824	0.106518	...	0.631741	255.749768	0.123106	-46.259673	7.870316	0
952	0.186149	-67.889444	0.027343	-7.277119	0.105508	...	0.645413	261.167041	0.122927	-46.177654	14.65699	0
953	0.186207	-67.949824	0.026665	-6.982705	0.106513	...	0.631842	255.795261	0.123116	-46.282577	16.070416	0
954	0.186582	-68.085028	0.024973	-6.658836	0.108393	...	0.611474	247.460571	0.123288	-46.331261	7.878267	0
955	0.186206	-67.94942	0.026669	-6.984159	0.106504	...	0.631921	255.826084	0.123153	-46.279575	16.070437	0

De un primer análisis se puede observar el siguiente patrón de los datos para el entrenamiento de la clasificación:

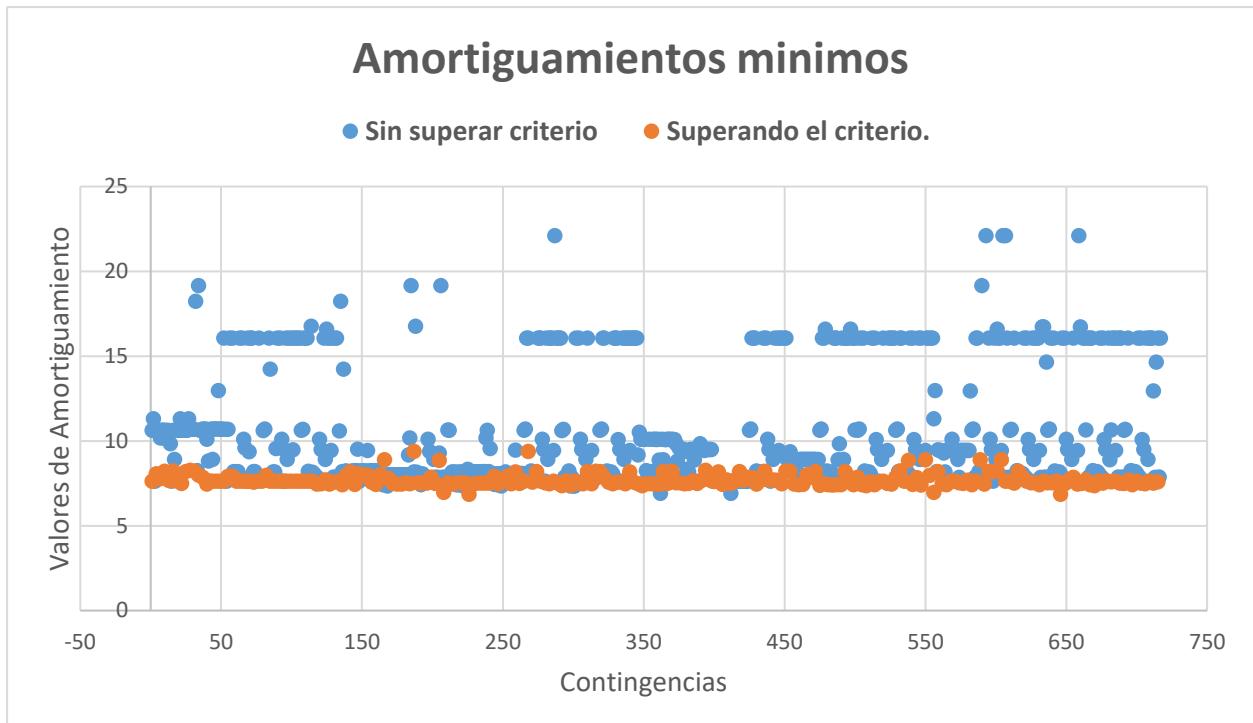


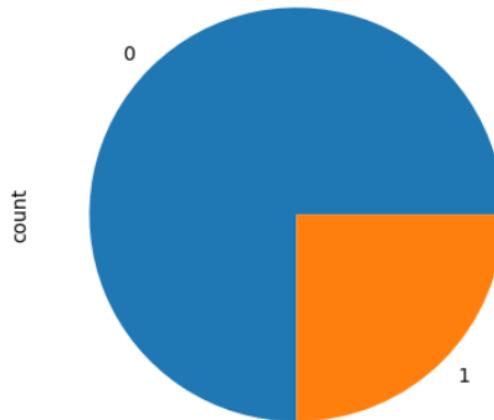
Figura 3.7 Patrón de comportamiento de los amortiguamientos mínimos.

3.3 Procesamiento de datos:

3.3.1 Resampling y entrenamiento de datos:

Mediante Jupyter Network se importa la data. Se hace una limpieza de datos previa, y se procesa la data.

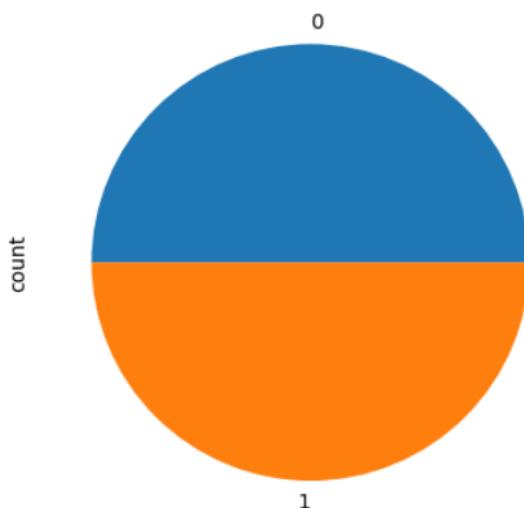
Se revisa los casos que superan el criterio respecto de los que no superan mediante un gráfico pie (valores de “1”):



```
Clasificacion  
0    717  
1    239  
Name: count, dtype: int64
```

Figura 3.8 Distribución de contingencias que superan o no el criterio.

Aplicamos la técnica de Re-sampling para equilibrar los datos y obtener un mejor entrenamiento:



```
Clasificacion  
0    717  
1    717  
Name: count, dtype: int64
```

Figura 3.9 Remuestreo de los datos obtenidos.

Debido a la diferencia entre orden de magnitudes de cada parámetro, se aplica una técnica llamada StandardScaler para tener los datos dentro de valores cercanos a la unidad, con fin de lograr un procesamiento de datos más eficiente resultando:

	Current, Magnitude/Terminal i in kA	Active Power/Terminal i in MW	Current, Magnitude/Terminal i in kA.1	Active Power/Terminal i in MW.1	Current, Magnitude/Terminal i in kA.2	Active Power/Terminal i in MW.2	u, Magnitude in p.u.	U, Angle in deg	Current, Magnitude/Terminal i in kA.3	P
0	0.291777	-0.197707	-0.258922	0.153507	0.423871	0.447349	1.108928	2.138293	0.255627	
1	0.291777	-0.197709	-0.258966	0.153474	0.423834	0.447325	1.103296	2.138550	0.255602	
2	0.291822	-0.197690	-0.258966	0.153115	0.423135	0.446897	1.077358	2.122033	0.255212	
3	0.291822	-0.197723	-0.259146	0.153456	0.423761	0.447361	1.089303	2.147458	0.255652	
4	0.291844	-0.197833	-0.259910	0.154303	0.425343	0.448614	1.097324	2.211074	0.256846	
...
1429	0.297478	-0.198870	-0.281470	0.138079	0.391853	0.436392	-1.797676	0.815740	0.247176	
1430	0.297297	-0.197323	-0.270645	0.124718	0.367932	0.417275	-1.870200	0.641794	0.228905	
1431	0.294107	-0.195722	-0.252409	0.124999	0.368447	0.411493	-0.401284	0.370656	0.222560	
1432	0.299107	-0.202247	-0.309049	0.163931	0.427036	0.474562	-2.979395	0.734790	0.285443	
1433	-0.234677	-0.169831	0.074995	0.089836	-0.230691	0.293707	-0.653840	-3.397190	-0.547731	

Figura 3.10 Data de entrenamiento previo al escalamiento.

	Current, Magnitude/Terminal i in kA	Active Power/Terminal i in MW	Current, Magnitude/Terminal i in kA.1	Active Power/Terminal i in MW.1	Current, Magnitude/Terminal i in kA.2	Active Power/Terminal i in MW.2	u, Magnitude in p.u.	U, Angle in deg	Current, Magnitude/Terminal i in kA.3	P
0	-4.071933	4.175621	0.439655	2.002074	0.130171	-0.072139	0.922348	0.306325	0.443499	
1	0.411793	-0.252382	-1.358218	0.879367	0.412869	0.428740	0.912732	0.364156	1.355179	
2	0.226930	-0.262681	0.613577	-0.980330	-3.615057	-3.718734	-0.305673	-0.921762	-0.971582	
3	0.284005	-0.182225	-0.225898	0.137497	0.376670	0.412017	0.905984	0.409405	0.205080	
4	0.286718	-0.188084	-0.267439	0.176384	0.449253	0.473698	0.157935	1.164396	0.262530	
...
1142	0.234345	-0.090378	-0.150962	0.062817	-3.615057	-3.718734	-1.775114	0.736755	0.217968	
1143	0.321501	-0.352301	0.588027	-0.915338	-0.720230	-0.304319	-0.926685	-1.045556	-0.930465	
1144	-0.259449	-0.156274	0.073761	0.096202	-0.244326	0.288582	-0.645281	-3.354672	-0.552624	
1145	-0.635514	-0.061936	-0.314339	0.270830	0.678435	-0.045723	0.359709	-0.682356	-0.527329	
1146	0.286718	-0.181690	-0.231214	0.125441	0.331497	0.397198	-0.391039	-0.285328	0.194679	

Figura 3.11 Data de entrenamiento después del escalamiento.

También se hace un cross validation para aprovechar los datos y hacer un buen entrenamiento.

3.3.2 Reducción por componentes principales:

Siendo que tenemos cerca de 125 variables, para lograr el entrenamiento del modelo en menor tiempo usaremos la técnica de componentes principales (PCA), resultando la contribución de cada parámetro en el siguiente cuadro:

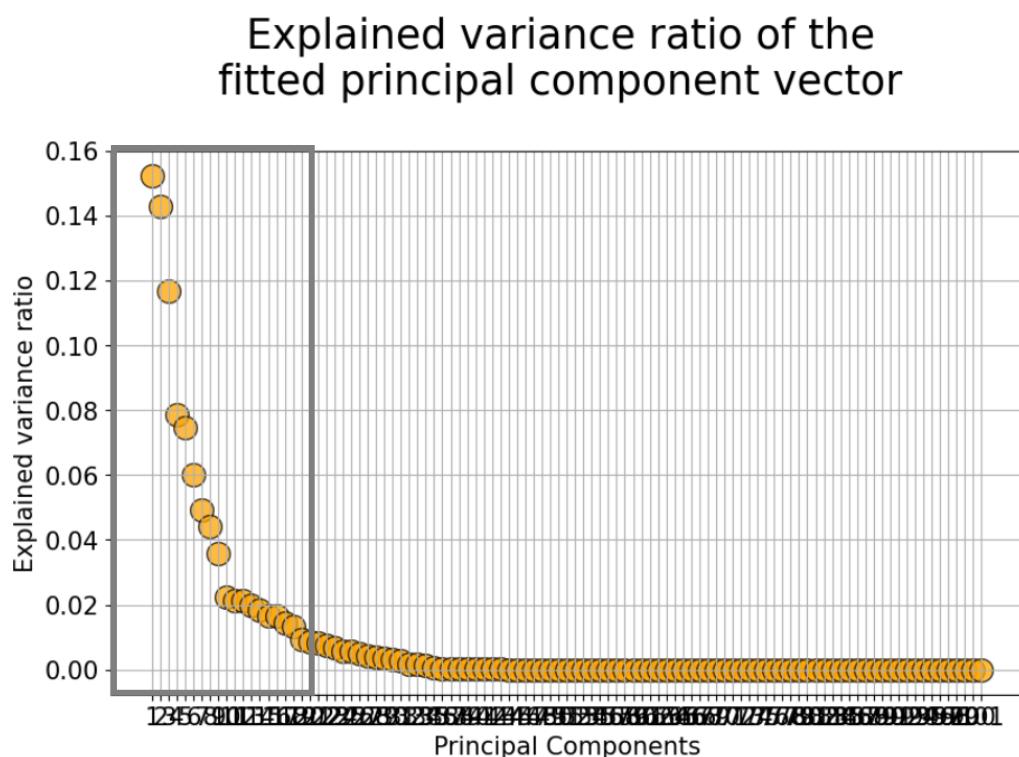


Figura 3.12 Método del codo aplicado a la data obtenida.

Determinando que la cantidad de componentes se pueden reducir de 102 a 20 variables significativas ($n=20$).

3.3.3 Entrenamiento y evaluación del modelo:

Haremos la separación de datos en data de entrenamiento y de prueba:

X_train, Xval, X_test, y_train, y_test

Donde se elegirá 20% para el tamaño de la data de prueba Xtest. El otro 80% se asignará a los datos de Xtrain y Xval. El Xval se consigue usando el 20% de la data de entrenamiento.

```
from sklearn.model_selection import train_test_split
val_size=0.2
X_train_scaled, X_val_scaled, y_train, y_val = train_test_split(X_train,y_train,test_size=val_size,random_state=42,shuffle=True)
```

Figura 3.13 División de la data entrenamiento, prueba y validación.

Se evaluarán los siguientes algoritmos: Random Forest, Logistic Regression y K nearest Neighbors. Siendo que la mejor forma de evaluar la utilidad de un modelo previamente entrenado es compararlo con otros algoritmos, hacemos una comparación entre ellos.

Tabla 5.5 Media y desviación de los datos.

	Media	Desviación estandar
RF	0.9946	0.0077
LR	0.9717	0.0172
KNN	0.9663	0.0210

Esto se muestra en la siguiente figura:

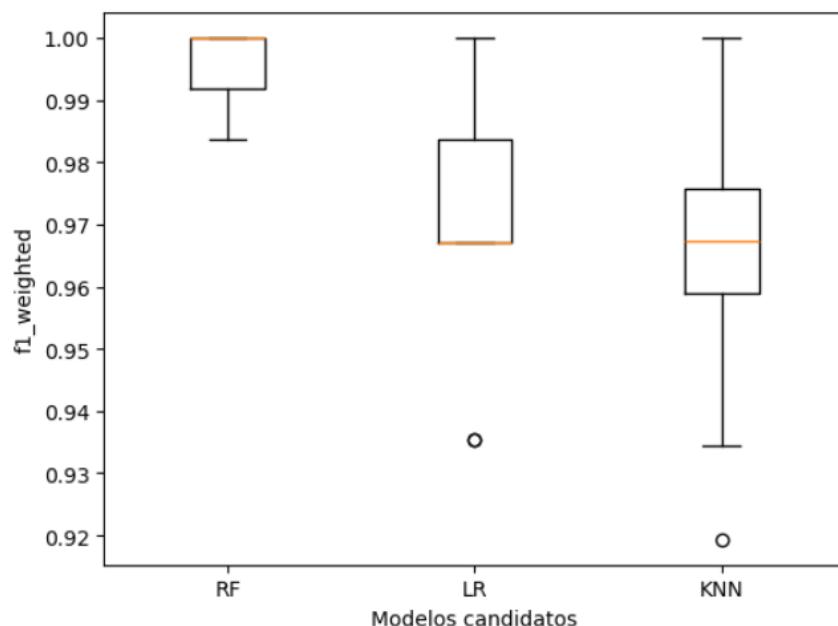


Figura 3.14 Comparación entre los modelos aplicados.

Debido al modelo de menor error y buena exactitud, *escogemos el modelo de Random Forest*.

Seguidamente estos resultados buscaremos optimizar el modelo mediante una mejor estimación de parámetros, para eso una técnica llamada GridSearchCV, la cual ejecuta a través de los diferentes parámetros y extrae los mejores valores y combinaciones de los mismos.

Para el Random Forest seleccionado, trabajamos sobre 2 de sus diferentes parámetros, obteniendo:

```
params = {
    'n_estimators':[200, 700],
    'max_features':['sqrt', 'log2']
}
```

Figura 3.15 Elección de parámetros de optimización.

Obteniendo la matriz de confusión:

	precision	recall	f1-score	support
0	0.95	0.98	0.97	144
1	0.98	0.95	0.96	143
accuracy			0.97	287
macro avg	0.97	0.97	0.97	287
weighted avg	0.97	0.97	0.97	287

Figura 3.16 Matriz de confusión obtenida.

De la matriz de confusión observamos que tenemos buenos valores de precision y recall, es decir que los casos se predicen vulnerables un 98% sobre las posibilidades de todas las predicciones de vulnerabilidad y un 95% para los casos que predijimos vulnerabilidad de los casos que realmente se tuvo vulnerabilidad.

Lo cual nos indica que el modelo tendrá una adecuada precision. Entonces podremos predecir con suficiente exactitud condiciones de falla en el sistema eléctrico a fin de toma de acciones para remediación.

3.3.4 Viabilidad técnica – económica

Para el traslado a la práctica se necesitan algunos elementos necesarios, tales como:

- ✓ PMUs (IEDs con la funcionalidad con sincronización de tiempo precisa).
- ✓ Canales de alta rapidez y bajas latencias.
- ✓ Enlaces por protocolos de comunicación hacia centro de control.
- ✓ PDCs.
- ✓ Servidores para procesamiento.

Los PMUs tomaran los datos del sistema eléctrico y los disponibilizaran a una tasa elegida de 60 muestras/ciclo para su reporte mediante protocolo IEEE C37.118-2014 (IEEE Standard for Synchrophasor Measurements for Power Systems) hacia centro de control.

La fibra óptica será el medio de transporte de datos (canal) para llevarla. Cabe indicar que se deben validar diversos parámetros de la fibra, siendo el principal la latencia, la cual es determinante para la evaluación en tiempo real del estado del sistema, previo procesamiento de la data recepcionada. Se requieren además ODFs y cajas de empalme.

Los PDCs tomarán los datos recepcionados de diferentes puntos de red y harán un ordenamiento y limpieza en el tiempo de cada uno de ellos de acuerdo a la estampa de tiempo registrada, dejando disponibles la data para procesamiento.

Los servidores serán los encargados de receptionar data y procesarla. Aquí se instalarán algoritmos de Machine learning y se obtendrá respuestas y/o resultados que harán indicación del estado del sistema.

Se puede resumir lo indicado en la siguiente figura:

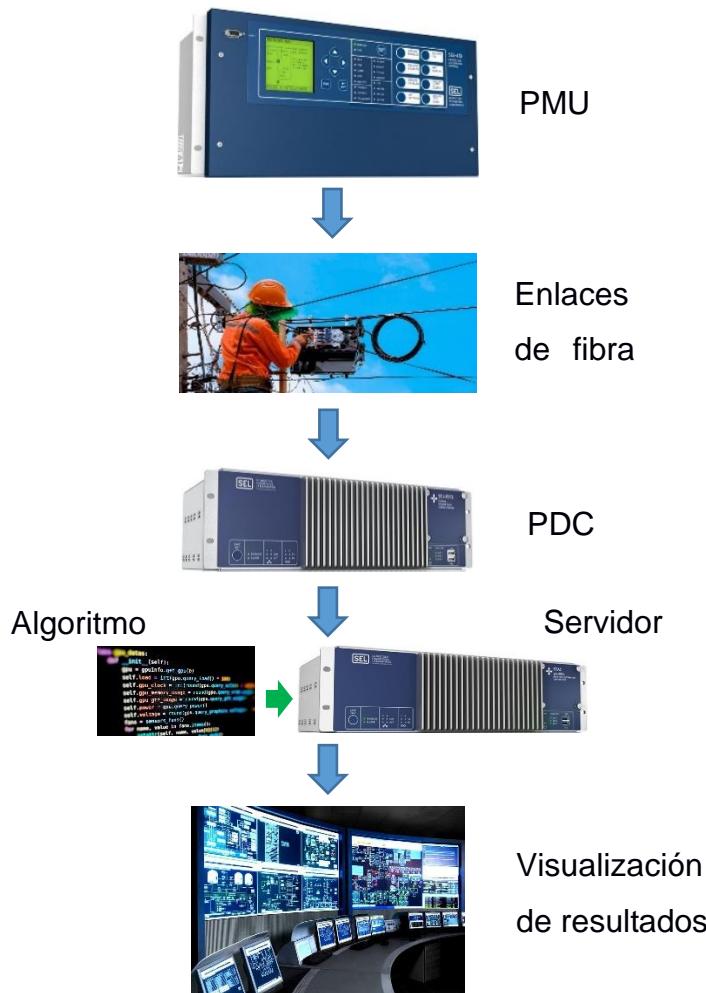


Figura 3.17 Estructura física para la evaluación de vulnerabilidad del sistema.

En cuanto a la factibilidad técnica, se detalla el siguiente cuadro con algunos costos aproximados:

Tabla 6.2 Esboce de costes asociados.

Equipo /IED	Precio (\$USD)
PMU - SEL451	7400
Fibra óptica /500m	5800
PDC - (SEL 3573/3555)	10500
Servidor SEL3555	10500
Comisionamiento	15000
<i>Total aprox.</i>	49200

Pero también se debe considerar que:

- Existen IEDs instalados en las subestaciones que ya tienen disponible la funcionalidad de PMU.
- La fibra óptica también se encuentra instalada en muchas de las instalaciones de los clientes propietarios de red.
- En algunas marcas instaladas de los actuales servidores y concentradores de datos ya se incorpora la funcionalidad de PDC si es así solicitada o con una actualización de firmware.

Siendo así que, para una empresa propietaria de la red, el cuadro podría actualizarse en un ahorro de hasta más del 50%, quedando prácticamente el desafío de presentar desarrollos de algoritmos y/o procesos sobre la infraestructura que ya se tiene.

CAPÍTULO 4

4. CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES:

- La evaluación rápida de la estabilidad oscilatoria del sistema peruano mediante la clasificación de data es factible y se puede aplicar en tiempo real para el sistema peruano usando la data del sistema Scada – WAMS.
- La evaluación de vulnerabilidad mediante análisis de estabilidad oscilatoria junto a las herramientas de aprendizaje automático proporciona diagnósticos confiables cuando se tiene un buen entrenamiento.
- Se evitan los grandes tiempos en corrida de flujos y análisis eléctricos con modelos debidamente entrenados para sistemas grandes como el sistema peruano, los cuales, para la capacidad de software y hardware utilizados en este trabajo, está en orden de los pocos segundos.
- Es posible la predicción de estados que tienden a ser anómalos mediante herramientas de Machine Learning aplicada para un modelo de un sistema eléctrico nacional.
- Un conjunto de variables de sistema puede ser considerada para la predicción de estados de falla siempre que presenten una alta precision en la evaluación del modelo. El prueba y error es necesario hasta que se obtenga un buen entrenamiento.

- El proceso de reducción de red del sistema es laborioso, sobretodo en sistemas grandes como el peruano. Mas este es un paso necesario previo al entrenamiento del modelo.
- Estos modelos están prestos para trabajar de mano del hardware y software actual de redes eléctricas. PMUs ahora están disponibles en los IEDs actuales.
- La alerta temprana se genera cuando el modelo reconoce y predice estados fuera de lo que se le ha parametrado.

4.2 RECOMENDACIONES:

- Realizar el enfoque del análisis a regiones específicas de la red mediante adecuados equivalentes thevenings. Esto ahorra mucho el tiempo invertido de generación de data del modelo.
- Se requiere hacer el análisis con data de estados de la red que se tienen de las mediciones diarias del centro de control. Esta data contiene unas características más reales del comportamiento en línea del sistema, mejorando la eficacia y predicción del modelo.
- Se debe realizar la metodología explicada para una data mejorada mediante la medición mixta que proviene de SCADA – PMU.

BIBLIOGRAFÍA

- [1] ISA - CENACE - COES - REP, «Informe de Estudios Interconexión Colombia – Ecuador – Perú», Medellin, Colombia, Estudio., oct. 2023. [En línea]. Disponible en: https://view.officeapps.live.com/op/view.aspx?src=https%3A%2F%2Fwww.coes.org.pe%2Fdataweb%2F2004%2Fdev%2Fcesi%2Finforme_final.doc&wdOrigin=BROWSELINK
- [2] «3_-_Equivalente_externo_el_Sistema_-_Inf_SPL-007_Interconexion_Ecuador_Peru.pdf». Accedido: 29 de noviembre de 2024. [En línea]. Disponible en: https://www.investinperu.pe/RepositorioAPS0/0/2/jer/LT_500KV_PIURA/3_-_Equivalente_externo_el_Sistema_-_Inf_SPL-007_Interconexion_Ecuador_Peru.pdf
- [3] COES, «Desconexión de las líneas L-2053/L-2054 (Cotaruse - Socabaya) de 220 kV», COES, Peru, Informe tecnico COES/D/DO/SEV/IT-098-2013, nov. 2013. [En línea]. Disponible en: <https://www.coes.org.pe/portal/browser/download?url=Post%20Operaci%C3%B3n/Analisis%20de%20Fallas/2013/769/EV-ITF-2013-769.pdf>
- [4] COES, «DESCONEXIÓN DEL REACTOR R-26 DE LA S.E. CHILCA 220 kV», COES, Peru, Informe tecnico EVENTO-017-2023, abr. 2023. [En línea]. Disponible en: <https://www.coes.org.pe/fileapp/home/download?url=Post%20Operaci%C3%B3n/Analisis%20de%20Fallas/2023/1624/EV-CTI-2023-1624.pdf>
- [5] J. Xie, I. Alvarez-Fernandez, y W. Sun, «A Review of Machine Learning Applications in Power System Resilience», en *2020 IEEE Power & Energy Society General Meeting (PESGM)*, Montreal, QC, Canada: IEEE, ago. 2020, pp. 1-5. doi: 10.1109/PESGM41954.2020.9282137.
- [6] H. Anand y M. Darayi, «Power Network Component Vulnerability Analysis: A Machine Learning Approach», *Procedia Comput. Sci.*, vol. 185, pp. 73-80, 2021, doi: 10.1016/j.procs.2021.05.008.
- [7] Z. Hou, J. Follum, P. Etingov, F. Tuffner, D. Kosterev, y G. Matthews, «Machine Learning of Factors Influencing Damping and Frequency of Dominant Inter-area Modes in the WECC Interconnect», en *2018 IEEE International Conference on*

- Probabilistic Methods Applied to Power Systems (PMAPS)*, Boise, ID: IEEE, jun. 2018, pp. 1-6. doi: 10.1109/PMAPS.2018.8440361.
- [8] J. C. Cepeda Campaña, «Evaluación de la Vulnerabilidad del Sistema Eléctrico de Potencia en Tiempo Real usando Tecnología de Medición Sincrofasorial», Universidad Nacional de San Juan, Argentina, 2014.
 - [9] COES, «Pagina web oficial de COES». [En línea]. Disponible en: <https://www.coes.org.pe/portal/>
 - [10] G. E. Ramirez Estrada, «Definición y Clasificación del Sistema de Potencia Estabilidad». 2016. [En línea]. Disponible en: <https://www.studocu.com/ec/document/universidad-tecnica-de-cotopaxi/sistemas-electricos-de-potencia/tarea-01-ramirez-estrada/36955858>
 - [11] «Definition and Classification of Power System Stability IEEE/CIGRE Joint Task Force on Stability Terms and Definitions», *IEEE Trans. Power Syst.*, vol. 19, n.º 3, pp. 1387-1401, ago. 2004, doi: 10.1109/TPWRS.2004.825981.
 - [12] DlgsILENT GmbH, «Power Factory User Manual». DlgsILENT GmbH, enero de 2021.
 - [13] COES, «Estudio de Estabilidad del SEIN 2022-2025», Lima, Estudio. EEG-013-2022, jul. 2022.