



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA

DEPARTAMENTO DE ELECTROTECNIA E INFORMÁTICA

INGENIERÍA EN INFORMÁTICA

Levantamiento de Fuentes de Información y Procesamiento de Datos e Implementación de un Modelo Predictivo de Fallas en Redes Eléctricas

Krithobal Riquelme Bustamante

Viña del Mar, Chile

Krithobal.riquelme@usm.cl

Gabriel Jara Bulnes

Profesor Guía

Resumen: En el presente trabajo se aborda la problemática de las fallas en redes eléctricas, con un enfoque particular en el sector de transmisión, el cual enfrenta riesgos constantes debido a factores como condiciones meteorológicas adversas y la falta de herramientas que permitan anticipar estos eventos. Estas limitaciones derivan en una gestión reactiva de los recursos ante la ocurrencia de fallas.

La propuesta se centra en desarrollar una herramienta predictiva basada en un modelo de red neuronal densamente conectada, cuyo objetivo es estimar la probabilidad de fallas eléctricas ocasionadas por condiciones climáticas. El procedimiento incluye la recopilación de fuentes de información, el procesamiento y limpieza de datos, y la posterior implementación del modelo predictivo.

Los resultados obtenidos muestran métricas prometedoras: una precisión que varía entre el 80 % y el 85 %, una pérdida que, aunque fluctúa, no desciende del 60 %, y un recall que oscila entre el 52 % y el 60 % en algunas iteraciones. Estos resultados destacan el potencial del modelo para contribuir a una gestión más eficiente y proactiva de los recursos en el sector eléctrico.

Palabras claves: Redes eléctricas, Modelo predictivo, fallas eléctricas, Extracción de datos



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): Memoria o trabajo de título; Tesis de Postgrado;

Título del trabajo: Levantamiento de Fuentes de Información y Procesamiento de Datos e implementación de un Modelo Predictivo de Fallas en Redes Eléctricas

Nombre del candidato(a): Kristhobal Benjamin Riquelme Bustamante

Carrera / Grado: Ingeniería en Informática

Campus: Viña del Mar; **Departamento:** Electrotecnia e Informática

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Gabriel Jara Bulnes, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL

El trabajo **NO contiene información que amerite confidencialidad** y puede ser publicado de inmediato en repositorio con acceso abierto.

El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (embargo) por:

6 meses; 12 meses; 2 años; 3 años; 5 años; 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 25/08/2025

; Firma:

Estudiante o Candidato(a):

Fecha: 21/08/2025

; Firma:



Contenido

1	Introducción	3
1.1	Descripción del Problema.....	4
1.2	Propuesta de Solución	4
1.3	Objetivo General	5
1.4	Objetivos Específicos	5
2	Marco Conceptual.....	5
2.1	Redes eléctricas:	5
2.2	Fallas Eléctricas.....	6
2.3	Predicción de Fallas	6
2.4	Modelo Predictivo	7
2.5	Redes Neuronales Aplicadas a Modelos Predictivos de Fallas	7
2.6	Minería de Datos en Predicción de Fallas.....	7
2.7	Técnicas de Balanceo: UnderSampling.....	8
2.8	Uso de APIs para Datos Meteorológicos	8
2.9	Inteligencia de Negocios (BI) para el Análisis de Redes Eléctricas	8
3	Metodología	9
3.1	Levantamiento de Información:	9
3.2	Extracción y Almacenamiento:.....	9
3.3	Procesamiento y Limpieza:	9
3.4	Desarrollo del Modelo:	9
3.5	Validación del Modelo:	10
4	Levantamiento y Procesamiento de datos.....	11
4.1	Identificación de Fuentes de Datos	11
4.2	Proceso de Extracción de Datos.....	12
4.3	Almacenamiento de Datos	17
4.4	Limpieza y Transformación de Datos	19
4.5	Descripción de los Datos Procesados	20
5	Implementación del Modelo Predictivo.....	20
5.1	Datos de Entrenamiento	21
5.2	Arquitectura.....	22
5.3	Hiperparámetros.....	23
5.4	Validación	25
6	Conclusión	26
7	Referencias.....	27



1 Introducción

Las redes eléctricas constituyen una infraestructura esencial para cualquier sociedad moderna, ya que su correcto funcionamiento garantiza un suministro de energía confiable a hogares, empresas, hospitales y otras áreas fundamentales. La electricidad es la base de la mayoría de las actividades productivas y sociales; cualquier interrupción en su flujo puede tener repercusiones significativas, como pérdidas económicas, disrupciones operativas y efectos sociales adversos.

Es importante destacar que los sistemas eléctricos están divididos en tres grandes etapas: generación, transmisión y distribución. La generación es el proceso mediante el cual se produce la energía en plantas o centrales eléctricas. La transmisión implica el transporte de esta energía a largas distancias mediante líneas de alta tensión, mientras que la distribución se encarga de llevar la electricidad desde las subestaciones hasta los consumidores finales. La interdependencia y complejidad de estos sistemas hacen que un fallo en cualquiera de estas etapas pueda desencadenar efectos en cascada que impacten grandes áreas, e incluso regiones enteras.

Bajo este contexto, las herramientas de análisis predictivo se posicionan en un apoyo clave para poder reducir los riesgos de cortes en sistemas de transmisión. Aunque no solucionan por completo el problema, pueden ser parte para poder anticipar posibles fallas y así lograr tomar medidas preventivas. Esto ya que estas herramientas se enfocan en detectar patrones y señales que podrían relacionarse con la falla y con esto permite que se actúe de una manera más rápida y eficaz.

Mediante una recopilación y análisis profundo de los datos que provienen de diversas fuentes como, por ejemplo: "Condiciones Meteorológicas", ayudan a las herramientas de análisis predictivo en este caso a modelos a que estimen una probabilidad de que ocurra una falla en sistemas de transmisión y así lograr gracias a estas herramientas, la empresa puede asignar de manera más eficiente sus recursos operativos, tales como personal técnico, equipos de mantenimiento, vehículos y materiales, logrando así una mejora significativa en su capacidad de respuesta ante posibles eventos.

Esta información tiene como objetivo presentar el desarrollo de una tesina que se centra en el procesamiento de datos y la implementación de un modelo predictivo de fallas en redes eléctricas. Este trabajo se enfoca en documentar el proceso, desde la recopilación y análisis de datos hasta la creación de un modelo que ayude a las empresas de transmisión a estimar la probabilidad de fallas en sistemas de transmisión y optimizar la asignación de recursos para su gestión.

Para el desarrollo de esta propuesta, este se distribuirá entre cinco capítulos, los cuales se detallarán a continuación:

- 1- **Introducción:** Se dará contexto de esta tesina desde la definición del problema, el planteamiento de objetivos entre otros.
- 2- **Marco Conceptual:** En este capítulo se abordarán los conceptos fundamentales que sustentan la investigación, proporcionando una base teórica sólida.
- 3- **Metodología de la tesina.**
- 4- **levantamiento y procesamiento de datos.**
- 5- **implementación del modelo predictivo.**
- 6- **Conclusiones:** Análisis de los resultados posteriormente a la integración y validación
- 7- **Referencias:** Literatura, sitios web y libros consultados en la elaboración de este documento.



1.1 Descripción del Problema

Uno de los mayores desafíos que se enfrentan las compañías de transmisión eléctrica es la gestión eficiente de los recursos como lo puede ser el desplazamiento de cuadrillas junto a su organización, la gestión del equipamiento necesario para la falla y la planificación de respuesta ante posibles fallas en los sistemas de transmisión y junto a una falta de un sistema que integre la información disponible de diversas fuentes de alimentación y asigne una probabilidad de una ocurrencia de una posible falla, limita a las empresas del rubro en la toma de decisiones.

Si bien actualmente existen ciertos indicadores que sugieren posibles fallos, la ausencia de modelos dificulta la preparación de los recursos y obligan a las empresas a una gestión reactiva lo que resulta en tiempos de respuestas prolongados, costos elevados de operativos y un uso ineficiente de los recursos disponibles (personal técnico, equipos de mantenimiento, movilización de cuadrilla, entre otros).

En este contexto surge **ElecPreTx**, una solución tecnológica que busca mejorar la gestión operativa mediante dos componentes clave:

1. **La visualización e integración de datos históricos y operacionales** en una plataforma amigable, que facilita la supervisión y análisis del estado de la red.
2. **El desarrollo de un modelo predictivo de fallas**, que permite estimar a una ocurrencia de eventos de riesgo y optimizar la asignación de recursos antes de que ocurran las fallas.

1.2 Propuesta de Solución

Frente a los desafíos que enfrentan las empresas de transmisión eléctrica en la gestión de fallas y uso eficiente de recursos, el proyecto **ElecPreTx** propone una solución integral que combina la visualización de datos históricos con la implementación de un modelo predictivo de fallas.

La propuesta se fundamenta en el desarrollo de un sistema que integre diferentes componentes:

- **Visualización de datos históricos** en una plataforma amigable, que facilita la supervisión y análisis del estado de la red.
- **Consolidación de fuentes de información**, como informes de fallas y datos meteorológicos.
- **Implementación de un modelo predictivo de fallas**, capaz de identificar patrones previos, para poder estimar una ocurrencia de falla entregando alertas con base probabilística.

La ventaja de este modelo radica en su capacidad para aprender de los datos históricos y actualizarse continuamente, mejorando su precisión con el tiempo. Al estimar una probabilidad de una falla se podrá realizar una optimización a la hora de planificación y distribución de los recursos, al hacer esto reducirá los tiempos de inactividad, minimizando también los costos operativos y mejorando en general la eficiencia de la gestión de los recursos. Aunque esto no garantiza predicción exacta, sino que ayuda a estimar una probabilidad de una falla este modelo contribuye consistente a que la empresa este mejor preparada ante una ocurrencia de alguna falla.

El enfoque integral de la propuesta de solución no solo se centra en la implementación de un modelo predictivo que pueda anticipar una ocurrencia a fallas en redes eléctricas, sino también en la implementación de un proceso exhaustivo de recolección y procesamiento de datos que asegure la efectividad del modelo.

En el marco de este proyecto, la presente tesis se enfocará en las primeras etapas del desarrollo de **ElecPreTx**, específicamente en el levantamiento de fuentes de información, procesamiento de datos y



la implementación inicial de un modelo predictivo de fallas en redes eléctricas y es importante considerar los siguientes aspectos:

- Identificación de Fuentes de Información.
- Extracción y procesamiento de Datos.
- Almacenamiento de Datos.
- Alimentación de Modelo Predictivo.
- Implementación de un Modelo Predictivo.
- Evaluación de resultados del modelo.

1.3 Objetivo General

Desarrollar un proceso eficiente de levantamiento, extracción y procesamiento de datos provenientes de diversas fuentes e implementar un modelo predictivo que permita estimar la probabilidad de ocurrencia de fallas en redes eléctricas.

1.4 Objetivos Específicos

- **Identificar y categorizar fuentes de información** relevantes para la predicción de fallas en redes eléctricas incluyendo datos meteorológicos, históricos de fallas.
- **Diseñar un proceso de extracción de datos** desde fuentes heterogéneas, garantizando la calidad, consistencia y disponibilidad de los datos.
- **Implementar técnicas de procesamiento de datos**, tales como imputación de valores faltantes y normalización, para preparar los datos de manera adecuada para su uso en el modelo predictivo.
- **Implementar, entrenar y validar un modelo predictivo** utilizando algoritmos de aprendizaje automático.

2 Marco Conceptual

El desarrollo de un modelo predictivo para anticipar fallas eléctricas en redes implica la comprensión y aplicación de varios conceptos clave, tanto en el ámbito de las redes eléctricas como en el análisis de datos.

2.1 Redes eléctricas:

Las redes eléctricas son infraestructura crítica que permiten la transmisión y distribución de suministro eléctrico, de esta forma tienen una importancia fundamental en el funcionamiento de la sociedad, de forma que provee energía a distintas estructuras (Hogares, Empresas, Hospitales, entre otros). La interrupción de alguna parte de la red puede desencadenar importantes consecuencias y riesgos, afectando tanto la estabilidad económica como la seguridad de las comunidades (WTS Energy, s.f.).

Las redes eléctricas se componen de tres partes esenciales y las cuales se detallarán a continuación:

- **Transmisión:** Involucra el transporte de electricidad a través de líneas de alta tensión. Este proceso se realiza a largas distancias y requiere subestaciones para transformar y regular el voltaje.
- **Distribución:** La cual reduce el voltaje y hace el suministro de electricidad a los consumidores finales (Hogares, empresas, servicios públicos, entre otros).

- **Generación:** Es el proceso mediante el cual se produce la electricidad en centrales eléctricas.

2.2 Fallas Eléctricas

Una **falla eléctrica** es cualquier interrupción, degradación o mal funcionamiento no planificado dentro del sistema eléctrico que compromete su operación normal. En el contexto de las redes de **transmisión**, estas fallas no solo afectan la continuidad del suministro, sino que pueden tener consecuencias operativas y económicas de gran magnitud, tales como pérdida de carga, desconexión de líneas críticas y activación de esquemas de protección que interrumpen el servicio a gran escala.

Uno de los principales desafíos que tienen las redes eléctricas es la **influencia de factores externos y ambientales**, los cuales muchas veces escapan al control directo de la operación de la red. Entre ellos destacan:

- **Condiciones meteorológicas extremas** (tormentas, lluvia intensa, nieve, hielo, viento fuerte).
- **Temperatura y humedad**, que afectan la resistencia de los materiales.
- **Presencia de vegetación o fauna cercana a las líneas.**
- **Contaminación en aisladores**, especialmente en zonas industriales o costeras.

Además de estos factores, la seguridad y el correcto funcionamiento de las líneas de transmisión pueden verse afectados por la degradación de los materiales debido a las condiciones ambientales, así como por la contaminación y presencia de fauna o vegetación cercana, que generan riesgos adicionales para la integridad del sistema (Electric Test, s.f.).

A pesar del avance en herramientas analíticas y modelos de aprendizaje automático, la predicción de fallas eléctricas presenta varios desafíos técnicos, entre los cuales destaca el desbalance natural de los datos. Es decir, en un conjunto de datos históricos, la cantidad de eventos de falla suele ser considerablemente menor que la de eventos sin falla, este fenómeno es conocido como problema de clases desbalanceadas. Adicionalmente, las fallas eléctricas pueden ser altamente contextuales, es decir, dependen no solo de condiciones actuales, sino también del historial de operación y de múltiples variables simultáneas.

2.3 Predicción de Fallas

El concepto de **predicción de fallas** implica el uso de datos históricos y en tiempo real para identificar patrones que indiquen la posible ocurrencia de una falla. Esta técnica se basa en el análisis predictivo, el cual utiliza diversos métodos y algoritmos para pronosticar eventos futuros con base en datos existentes.

En el contexto de redes eléctricas, la predicción de fallas busca anticipar eventos basados en la evaluación de las condiciones operativas y meteorológicas, así como en datos históricos de fallas y mantenimiento. Los modelos predictivos permiten a las empresas de transmisión eléctrica identificar las zonas o componentes con mayor riesgo de falla, facilitando la planificación preventiva y la optimización de la asignación de recursos, tales como personal técnico, equipos de mantenimiento y movilización de cuadrillas. De esta forma, los modelos predictivos se han convertido en herramientas esenciales para la gestión proactiva de las redes eléctricas, contribuyendo a la continuidad del suministro y a la reducción de impactos económicos y sociales.

2.4 Modelo Predictivo

El modelo predictivo es el núcleo del proyecto, y se refiere a un sistema basado en datos que utiliza técnicas estadísticas y algoritmos de aprendizaje automático para prever la ocurrencia de eventos futuros, en este caso, fallas en redes eléctricas. Estos modelos buscan identificar patrones y correlaciones en los datos históricos, como condiciones meteorológicas, registros de fallas previas y otros factores relevantes, para estimar la probabilidad de fallas futuras.

2.5 Redes Neuronales Aplicadas a Modelos Predictivos de Fallas

Las redes neuronales artificiales (RNA) son una de las herramientas más potentes dentro del aprendizaje automático para la predicción de fallas, debido a su capacidad para aprender patrones complejos y no lineales a partir de grandes volúmenes de datos. Estas redes están inspiradas en el funcionamiento del cerebro humano, y consisten en capas de nodos interconectados que transforman datos de entrada en salidas útiles mediante funciones de activación y aprendizaje supervisado o no supervisado (BBVA, 2024).

En el contexto de redes eléctricas, las redes neuronales se utilizan para predecir fallas mediante la correlación de datos históricos, condiciones meteorológicas, registros operativos y otras variables relevantes. Estas predicciones permiten anticiparse a eventos críticos y facilitar una gestión proactiva de los recursos técnicos.

Una de las arquitecturas más comunes es la **Red Neuronal Densamente Conectada** en este tipo de red, cada neurona de una capa está conectada con todas las neuronas de la siguiente capa, lo que permite una mayor propagación y reutilización de la información.

Sin embargo, existen otras arquitecturas que pueden ser más adecuadas dependiendo del tipo de datos disponibles:

- **Redes Neuronales Recurrentes (RNN):** Son eficaces para datos secuenciales o temporales, como series de tiempo, que son comunes en registros de sensores y condiciones operativas de redes eléctricas.
- **Redes Convolucionales (CNN):** Aunque son más utilizadas en procesamiento de imágenes, también han sido aplicadas en predicción de fallas.

2.6 Minería de Datos en Predicción de Fallas

El desarrollo de modelos predictivos en el contexto de redes eléctricas se enmarca en el ciclo de minería de datos, el cual comprende una serie de etapas fundamentales que permiten extraer conocimiento útil a partir de grandes volúmenes de datos. Este ciclo no solo aporta el sustento metodológico del proyecto, sino que también justifica la aplicación de técnicas específicas como el preprocesamiento, el balanceo de clases y el modelado con redes neuronales.

Etapas del Ciclo de Minería de Datos:

- **Comprensión del negocio:**
Esta etapa busca entender el problema desde la perspectiva operativa. En este caso, se trata de anticipar fallas en redes de transmisión eléctrica para optimizar la gestión de recursos y reducir impactos operativos.
- **Comprensión de los datos:**
Consiste en recolectar, explorar y analizar los datos disponibles, como registros de fallas, condiciones meteorológicas, características de líneas de transmisión, entre otros. Aquí se identifican inconsistencias, valores faltantes y relaciones potenciales entre variables.

- **Preparación de los datos (Preprocesamiento):**
Esta fase es crucial para mejorar el rendimiento del modelo. Incluye la limpieza de datos, normalización, codificación de variables, selección de características relevantes y, en este caso, tratamiento del desbalance de clases, ya que las fallas suelen ser eventos raros frente a los muchos registros de operación normal.
- **Modelado:**
En esta etapa se aplican algoritmos de aprendizaje automático, como redes neuronales, para construir modelos capaces de predecir la ocurrencia de fallas. La selección del modelo depende del tipo y estructura de los datos.
- **Evaluación:**
En esta etapa se evalúa el rendimiento del modelo utilizando métricas adecuadas para problemas de clasificación, especialmente en contextos donde existe un desbalance entre las clases.

2.7 Técnicas de Balanceo: UnderSampling

El UnderSampling es una técnica utilizada en problemas de clasificación, especialmente cuando se enfrenta a un conjunto de datos desbalanceado, donde una clase está significativamente sobrerrepresentada en comparación con las demás.

Dentro de las técnicas que puede haber en el Undersampling algunas son:

- **Edited Nearest Neighbours (ENN)**

El Edited Nearest Neighbours (ENN) es otra técnica para tratar el desbalance de clases. Funciona eliminando muestras de la clase mayoritaria.

- **Random Undersampling (RAS)**

Esta técnica consiste en seleccionar y descartar aleatoriamente las muestras de la clase mayoritaria para lograr un equilibrio en los datos.

2.8 Uso de APIs para Datos Meteorológicos

En la estimación de fallas eléctricas, los datos meteorológicos son fundamentales, ya que las condiciones climáticas extremas tienen un impacto directo sobre la infraestructura eléctrica, especialmente en las redes de transmisión. Fenómenos como lluvia intensa, viento fuerte, humedad elevada o temperaturas extremas pueden deteriorar componentes, inducir descargas eléctricas o generar condiciones propensas a fallas.

Para incorporar esta información al modelo predictivo, se emplearán servicios de terceros que proporcionan datos meteorológicos a través de **APIs**.

2.9 Inteligencia de Negocios (BI) para el Análisis de Redes Eléctricas

Estas Herramientas con respecto a la inteligencia de negocios son parte fundamental para la consolidación y análisis de grandes volúmenes de datos con respecto a los sistemas de transmisión, estas herramientas permiten transformar los datos en información accionable.

La implementación de BI en la gestión de redes eléctricas ayuda a una toma de decisiones más informada, permitiendo que las empresas de transmisión eléctrica optimicen la planificación de recursos y respondan positivamente ante los riesgos.



Proceso ETL (Extracción, Transformación y Carga)

El proceso de extracción, transformación y carga (ETL) es una práctica fundamental con relación a lo que es la inteligencia de negocios (BI) el cual este proceso consiste en garantizar que la información sea lo más consistente posible y a su vez es clave para el desarrollo de herramientas de análisis predictivos a continuación se explicaran las fases que conlleva este proceso:

- **Extracción:** Obtención de datos de diferentes fuentes, como registros históricos de fallas eléctricas y servicios externos como APIs meteorológicas.
- **Transformación:** Adaptación y procesamiento de los datos para asegurar su integridad y formato adecuado. Esto incluye la limpieza, normalización, entre otros.
- **Carga:** Almacenamiento de los datos procesados en almacenes de datos diseñados para consultas y análisis rápidos.

3 Metodología

La metodología por seguir en esta tesina se va a distribuir en las siguientes etapas:

3.1 Levantamiento de Información:

El primer paso es la identificación y recopilación de las fuentes de información relevantes para el proyecto. Esto incluye la obtención de datos históricos sobre fallas en la red eléctrica y registros de condiciones meteorológicas que puedan influir en estas fallas. Para posteriormente asegurar que el modelo se alimente con la calidad de más alta de los datos obtenidos.

3.2 Extracción y Almacenamiento:

Se diseñará un script robusto de extracción de datos que permita obtener la información de forma automatizada y eficiente. Una vez extraídos, los datos se organizarán y almacenarán en una instancia de almacenamiento optimizada para el acceso rápido y seguro. Este proceso garantiza que los datos estén disponibles para su posterior procesamiento y análisis, preservando su integridad y estructura.

3.3 Procesamiento y Limpieza:

La fase de procesamiento y limpieza de datos implica transformar los datos brutos en un formato adecuado para el uso en el modelo predictivo. Se llevarán a cabo procesos de depuración para eliminar inconsistencias, corregir valores erróneos y completar datos faltantes.

3.4 Desarrollo del Modelo:

La selección del algoritmo de aprendizaje automático es un paso clave para el éxito del proyecto. Se evaluarán diferentes opciones basadas en su capacidad de manejar datos complejos y proporcionar resultados consistentes. Una vez seleccionado el algoritmo, el modelo se entrenará con los datos previamente procesados, ajustando los parámetros para mejorar su capacidad de estimar la probabilidad de fallas en la red.



3.5 Validación del Modelo:

Pruebas de rendimiento del modelo para garantizar su precisión y efectividad, para medir cómo responde el modelo ante diferentes escenarios. Esta etapa permitirá identificar áreas de mejora y realizar ajustes necesarios antes de su implementación en un entorno real.



4 Levantamiento y Procesamiento de datos

La calidad y relevancia de los datos obtenidos son esenciales, porque son de lo que depende en mayor parte el éxito de un modelo predictivo. Para esto en esta sección, se describirán en detalle los pasos a realizar para el correcto funcionamiento de este modelo predictivo de fallas en redes de transmisión eléctrica, se verán desde identificar, extraer y procesar los datos necesarios para implementar en dicho modelo.

4.1 Identificación de Fuentes de Datos

En esta primera instancia consiste en identificar y categorizar las fuentes de información relevantes para el modelo, las fuentes pueden ser categorizadas de dos maneras:

1. Fuentes Internas

Las fuentes internas son datos generados y recopilados por una empresa u organización, estas fuentes obtenidas son exclusivas de dichas empresas y suelen incluir informaciones específicas de sus procesos realizados.

2. Fuentes Externas

Las fuentes externas son datos que se obtienen de servicios o entidades externos a la empresa o la organización, los cuales pueden proporcionar una información más amplia, incluyendo variables que las organizaciones no pueden medir directamente.

Dado este contexto de la categorización de las fuentes de información las fuentes que se identificaron para proporcionar datos al modelo predictivos fueron dos fuentes de información externas y que categorizaron de la siguiente manera.

1. Informes de Historial de fallas

Con Respecto a la fuente de información para obtener el historial de fallas, se ha seleccionado el **Coordinador Eléctrico Nacional (CEN)** el cual es el responsable de supervisar y coordinar la operación del sistema eléctrico, dentro de sus funciones el **CEN** tiene un registro de las fallas ocurridas a nivel nacional que ocurren en las redes de transmisión y distribución.

Dentro de los datos útiles que nos puede ofrecer de los informes de fallas el **CEN** pueden ser:

- **Fecha y hora** de la ocurrencia de una falla.
- **Ubicación** de la falla.
- **Causa** que describe el motivo de la falla.
- **Tensión nominal** de la instalación.

Donde la calidad y relevancia de esta información proporcionada de parte del **CEN** nos aseguran una representación de la información de forma precisa de lo que ocurrió en dicha estancia.

2. Historial de Datos Meteorológicos



Para este proyecto, fue necesario acceder a información meteorológica tanto histórica como en tiempo real, con el fin de integrar esta variable como factor de análisis en la predicción de fallas eléctricas. Las condiciones climáticas extremas, como fuertes vientos, precipitaciones intensas o altas temperaturas, son factores que influyen directamente en el comportamiento de las redes eléctricas y pueden estar correlacionados con fallas.

Tras un análisis comparativo entre diferentes fuentes de datos meteorológicos, se consideraron las siguientes alternativas:

- **VisualCrossing Weather** se posiciona como una solución integral con acceso vía API que incluye datos históricos de hasta 50 años, datos en tiempo real y pronósticos de hasta 15 días, ofreciendo alta cobertura geográfica.
- **OpenWeatherMap** ofrece una plataforma completa con acceso vía API que proporciona datos meteorológicos actuales, pronósticos de hasta 16 días y datos históricos de hasta 40 años.
- **Meteosource** ofrece una solución robusta con acceso vía API que incluye datos históricos de hasta 20 años completos junto a datos en tiempo real y pronósticos de hasta 5 días.

Si bien las fuentes de **OpenWeatherMap** y **VisualCrossing** son prácticamente iguales en términos de recursos ya sean en datos históricos o en el límite de llamadas diarias se decidió usar **VisualCrossing** ya que ofrecía una leve mejora de en a cantidad de datos (proporciona 50 años de datos históricos en cambio la otra API proporcionaba 40 años), pero con la diferencia que **OpenWeatherMap** solo lo permite en un plan pagado en diferencia de VisualCrossing que si se puede usar en su versión gratuita.

4.2 Proceso de Extracción de Datos

Una vez identificados y seleccionados las fuentes datos más relevantes para disponer al modelo, el siguiente paso fue desarrollar el proceso de extracción de datos de dichas fuentes. Para explicar lo realizado se dividirá en dos partes:

- **Extracción de Informes de Fallas (CEN)**

Los informes de fallas que nos proporciona el Coordinador Eléctrico Nacional (CEN) están disponibles en formato PDF, los cuales incluyen desde descripciones textuales, tablas o imágenes para esto el proceso de extracción para estos datos se dividió en dos partes: La primera consta de realizar un análisis manual del PDF para identificar los datos más relevantes y la segunda se enfoca en la extracción de los datos mediante scripts de Python.

La primera parte consistió en un análisis manual de los informes, donde se permitió identificar y seleccionar las secciones del documento que contenían la información más relevante, a continuación, se mostrarán los datos seleccionados y posteriormente algunos anexos para demostrarlos visualmente:

- **Fecha:** Fecha específica en que ocurrió la falla.
- **Hora:** Hora exacta del evento.
- **Nombre de la Instalación:** Identificación del componente o infraestructura afectada.
- **Tensión Nominal:** Nivel de tensión eléctrica asociado a la instalación.
- **Propietario:** Entidad responsable de la instalación afectada.
- **Origen y Causa:** Razón específica del fallo.

- **Comuna Afectada:** Ubicación geográfica de la falla.

Estudio para análisis de falla EAF 286/2024
"Falla en línea 2x44 kV FFCC Los Andes - Juncal N°2"
Fecha de Emisión: 12-07-2024

1. Descripción pormenorizada de la perturbación

a. Fecha y Hora de la falla

Fecha	21/06/2024
Hora	23:58
Consumos desconectados (MW)	0.44
Demanda previa del sistema (MW)	10805.06
Porcentaje de desconexión	0.004%
Calificación Apagón	No aplica (porcentaje de desconexión < 10%)

b. Identificación instalación afectada

Nombre de la instalación	Circuito N°2 de la línea 2x44 kV FFCC Los Andes - Juncal / LT027-----T0057
Tipo de instalación	Línea
Tensión nominal	44 kV
Segmento	Transmisión Zonal
Propietario instalación afectada	Chiquinta Transmisión S.A.
RUT	77.402.187-6
Representante Legal	Francisco Mualim Tietz
Dirección	Av. Argentina N°1, piso 9, Valparaíso, Región de Valparaíso.

Imagen 1 "Informe de Falla pagina 1"

d.1 Origen y causa de la falla

Se produjo la desconexión forzada del circuito N°2 de la línea 2x44 kV FFCC Los Andes - Juncal, por operación de su protección de distancia en zona 2 del paño F4 de S/E Los Andes (FFCC), ocasionado por la pérdida de aislación en el vano comprendido entre las estructuras N°127 a N°128.

La causa de la falla es atribuida a una pérdida de aislación provocada por el frente de mal tiempo (nieve) que afectaba a la zona.

Imagen 2 "Informe de Falla pagina 2"

h. Comuna donde se presenta la falla

5301: Los Andes.

Imagen 3" Informe de Falla pagina 3"

Una vez ya identificados los datos a extraer podemos empezar con la siguiente parte que se relaciona con la extracción de datos, donde se implementaron scripts en Python utilizando la biblioteca de PDFPlumber.

- **PDFPlumber**

Esta Biblioteca fue diseñada para la extracción de textos, tablas y entre otros elementos de archivos PDF y es especialmente útil para cuando se trabaja con documentos PDF que contienen datos estructurados como lo son los informes de fallas eléctricas.

A continuación, se mostrarán algunos de los scripts de Python que se utilizaron para la extracción de datos.



```
def formatear_filas_tabla(filas):
    lista_formateada = []
    for fila in filas:
        for celda in fila:
            lista_formateada.append(celda.strip())
    return lista_formateada

def extraer_datos_tabla(pagina, titulo, tabla_indice):
    tablas = pagina.extract_tables()
    if len(tablas) > tabla_indice:
        tabla = tablas[tabla_indice]
        indices = indices_filas.get(titulo, [])
        filas_seleccionadas = [tabla[i] for i in indices if i < len(tabla)]
        return formatear_filas_tabla(filas_seleccionadas)
    return []
```

Código Python N°1: Funciones para extracción de datos desde tablas en PDF

Tal como se ve en los dos extractos de código, se está realizando algunas funciones para extraer los datos de tablas que se encuentran en los informes PDF, la primera función llamada `formatear_filas_tabla` la cual se encarga de formatear una lista de filas extraídas, eliminando los espacios en blanco al inicio y final de cada celda de la tabla. La segunda función llamada `extraer_datos_tabla` está encargada de extraer los datos de una tabla específica en una página del documento identificando mediante un título ya definido y haciendo uso también de la primera función para limpiar y formatear las filas seleccionadas.

```
def extraer_datos_parrafo(pagina, titulo):
    texto = pagina.extract_text()
    parrafos = texto.split('\n')
    texto_parrafo = ""
    titulo_encontrado = False
    texto_acumulado = set()

    for parrafo in parrafos:
        # Buscar variantes del titulo en el párrafo
        for variante in titulos_variantes[titulo]:
            if variante in parrafo:
                titulo_encontrado = True

    if titulo_encontrado:
        # Agregar párrafos relevantes al texto acumulado
        if parrafo.strip() == "" or parrafo.strip().endswith("."):
            if parrafo.strip() not in texto_acumulado:
                texto_parrafo += " " + parrafo.strip()
                texto_acumulado.add(parrafo.strip())

    return texto_parrafo.strip() # Devolver el texto acumulado
```

Código Python N°2: Funcion para extracción de párrafos en PDF



La función que se hace mención llamada `extraer_datos_parrafo` se encarga extraer y acumular texto relevante de los párrafos de una página de un documento PDF, buscando información relacionada con un título específico.

- **Extracción de datos Meteorológicos (VisualCrossing)**

Las Condiciones Climáticas que nos brinda la herramienta API de VisualCrossing están disponibles en diferentes formatos (JSON, CSV y Excel) se seleccionó el formato de JSON para obtener los datos y se hizo una configuración para que las solicitudes a la herramienta fueran por HTTP.

Dentro de los datos relevantes para el modelo se encuentran cinco de ellos los cuales son:

- Temperatura (°C)
- Humedad (%)
- Precipitación (mm)
- Precipitación del Día (mm)
- Velocidad del Viento (km/h)
- Dirección del Viento (°)

Una vez que han identificados y seleccionados los datos empieza el proceso de extracción, pero para esto se realiza de una solicitud vía HTTP la cual se tiene que describir de la siguiente manera:

```
"https://weather.visualcrossing.com/VisualCrossingWebServices/rest/services/timeline/{ubicacion}/{fecha_hora}?key={API_KEY}&include=current&elements=temp,humidity,precip,windspeed,winddir&unitGroup=metric"
```

Enlace URL N°1: Para vista de datos meteorológicos

Donde a la solicitud HTTP se integran la ubicación, fecha y hora que nos brinda el CEN, la KEY de la herramienta API, la unidad de medida que en este caso es métrica y los datos seleccionados.

Si se hace ingreso a esta solicitud HTTP nos mostrara el siguiente script de JSON para este ejemplo se seleccionaron los datos esenciales ya mencionados, junto a la fecha 10/11/2024 a las 15:00 y la ubicación Viña del Mar, Chile:

```
{
  "queryCost": 1,
  "latitude": -33.0239,
  "longitude": -71.5522,
  "resolvedAddress": "Viña del Mar, Valparaíso, República de Chile",
  "address": "VinadelMar,Chile",
  "timezone": "America/Santiago",
  "tzoffset": -3,
  "days": [
    {
      "temp": 16.8,
      "humidity": 68,
      "precip": 0,
      "windspeed": 31.7,
      "winddir": 232
    }
  ]
}
```



```
    }  
  ],  
  "currentConditions": {  
    "temp": 23,  
    "humidity": 46.9,  
    "precip": 0,  
    "windspeed": 26,  
    "winddir": 250  
  }  
}
```

Código JSON N°1: Solicitud de las condiciones climáticas

En el extracto de código que se visualiza se ve separado en dos partes las condiciones climáticas para especificar más a fondo del contenido, las condiciones climáticas que se encuentran después de la palabra claves “days” son los promedios de las condiciones climáticas del día completo, en cambio las condiciones climáticas que aparecen después de “currentConditions” son las condiciones climáticas de la hora especificada.

Una vez que se obtuvieron los datos vía solicitud HTTP se realizan scripts de Python utilizando las bibliotecas de Requests, Datetime y Time.

- **Requests**

La biblioteca “**Requests**” permite enviar solicitudes HTTP de manera sencilla y eficiente. Esto incluye solicitudes GET, POST, PUT, DELETE, entre otras, como también manejar las solicitudes de HTTP para poder acceder al contenido de la respuesta.

- **Datetime**

La biblioteca “**Datetime**” que permite trabajar con fechas y horas y dentro de sus funcionalidades se encuentran la creación, manipulación y formateos con relación a las fechas y horas.

- **Time**

La biblioteca “**Time**” permite trabajar con el tiempo proporcionando funciones para pausar la ejecución, medir intervalos y formatear tiempos.

A continuación, se mostrarán algunos de los scripts de Python utilizados para la extracción de los datos meteorológicos junto a una breve explicación de lo que está sucediendo en ese extracto de código:

```
# Formatear la fecha  
fecha_dt = datetime.strptime(fecha, "%d/%m/%Y")  
fechaF = fecha_dt.strftime("%Y-%m-%d")  
  
# Redondear la hora  
hora_dt = datetime.strptime(hora, "%H:%M")  
if hora_dt.minute < 30:  
    hora_redondeada = hora_dt.replace(minute=0, second=0)  
else:  
    nueva_hora = hora_dt.hour + 1 if hora_dt.hour < 23 else 0
```



```
hora_redondeada = hora_dt.replace(hour=nueva_hora, minute=0,  
second=0)  
  
# Formatear la hora como HH:00:00  
horaf = hora_redondeada.strftime("%H:%M:%S")
```

Código Python N°3: Proceso para formatear la fecha y hora de los PDF

Este extracto de código se encarga del formateo y redondeo de las fechas y horas para así ajustarlo al formato requerido por la solicitud HTTP.

```
while reintentos < max_reintentos:  
    url=f"https://weather.visualcrossing.com/VisualCrossingWebServices/rest/services/ti  
meline/{ubicacion}/{fecha_hora_combinada}?key={API_KEY}&include=current&elements=tem  
p,humidity,precip,windspeed,winddir&unitGroup=metric"  
    response = requests.get(url)  
  
    if response.status_code == 200:  
        try:  
            data = response.json()  
            break # Salir del bucle si la solicitud fue exitosa  
        except ValueError:  
            print("Error al decodificar la respuesta JSON.")  
            data = {}  
            break  
    elif response.status_code == 429:  
        print("Demasiadas solicitudes. Esperando antes de reintentar...")  
        time.sleep(60)  
        reintentos += 1  
    else:  
        print(f"Error en la solicitud: {response.status_code}. Reintentando...")  
        reintentos += 1
```

Código Python N°4: Proceso para realizar solicitud a la API

En el extracto de código que se está mostrando se puede ver utilizando las bibliotecas de requests y time se hace la solicitud HTTP.

4.3 Almacenamiento de Datos

Para el almacenamiento de los datos recolectados, se opta por almacenarlos en un archivo formato Excel debido a su facilidad al momento de su manejo y a su vez con la compatibilidad que tiene con múltiples herramientas de análisis.

Una vez que ya se obtuvieron los datos a recolectar se realiza un script de Python para almacenarlos en un archivo Excel a continuación se detallaran los datos recolectados y posteriormente el script utilizado para el almacenamiento de dichos de datos.

- **Fecha:** Fecha específica en que ocurrió la falla.
- **Hora:** Hora exacta del evento.



- **Nombre de la Instalación:** Identificación del componente o infraestructura afectada.
- **Tensión Nominal:** Nivel de tensión eléctrica asociado a la instalación.
- **Propietario:** Entidad responsable de la instalación afectada.
- **Origen y Causa:** Razón específica del fallo.
- **Comuna Afectada:** Ubicación geográfica de la falla.
- **Temperatura (°C).**
- **Humedad (%).**
- **Precipitación (mm).**
- **Precipitación del Día (mm).**
- **Velocidad del Viento (km/h).**
- **Dirección del Viento (°).**

Para el script de Python que se encarga de almacenar los datos recolectados se utiliza las siguientes bibliotecas “os” y “pandas” las cuales serán detalladas a continuación:

- **Os**

La biblioteca “os” se encarga de cualquier tarea que requiera una interacción con el sistema operativo, permitiendo de este modo realizar operación de archivos y directorios.

- **Pandas**

La biblioteca “pandas” ofrece funciones que permiten el análisis, manipulación y visualización de datos de manera eficiente.

Con el contexto de las bibliotecas utilizadas para el almacenamiento se mostrará una parte del script realizado:

```
resultado = {
    "Fecha":
    "Hora":
    "Nombre de la instalación":
    "Tensión nominal":
    "Propietario instalación afectada":
    "Causa" :
    "Comuna":
    "Temperatura":
    "Humedad":
    "Precipitación":
    "Velocidad del viento":
    "Dirección del viento":
    "Precipitaciones del día" :
}
# Nombre del archivo Excel donde se guardarán los resultados
nombre_excel = "resultados_Recoleccion.xlsx"

# Verificar si el archivo ya existe
if os.path.exists(nombre_excel):
    df_existente = pd.read_excel(nombre_excel)
    df_nuevo = pd.DataFrame([resultado])
    df_actualizado = pd.concat([df_existente, df_nuevo], ignore_index=True)
else:
```

```
# Si el archivo no existe, se crear un nuevo DataFrame con el resultado
df_actualizado = pd.DataFrame([resultado])

df_actualizado.to_excel(nombre_excel, index=False)
```

Código Python N°5: Proceso para almacenar los datos recolectados

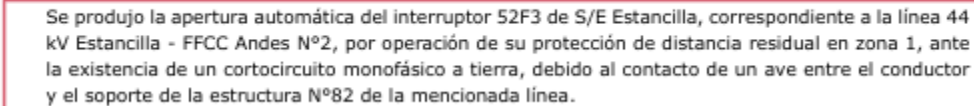
Funcionamiento de este script se basa en la creación de un archivo Excel o en caso de existencia la actualización de un archivo Excel mediante la biblioteca de “os” y la inserción de los datos mediante la biblioteca de “pandas”.

4.4 Limpieza y Transformación de Datos

Para el proceso de limpieza y transformación de los datos se tuvieron que analizar los datos recolectados y a partir de eso determinar que partes o componentes de un dato son necesarios, para ello se mostrarán los ejemplos de limpieza y transformación de los datos, el primero el cual se hizo una transformación del dato donde hubo una eliminación de partes irrelevantes y el segundo donde hubo una eliminación de partes irrelevantes a continuación, se mostrara visualmente los datos afectados junto a un extracto de código:

1. Lo que está sucediendo en este caso es que a través de la recolección de la causa el dato se compone de todo el párrafo lo cual no es muy efectivo para el modelo predictivo, lo cual conlleva realizar una categorización del párrafo y a través de palabras claves poder identificar la causa de la falla para este ejemplo se podrá decir que entra en una categorización de la causa de la falla fue debido a un ave.

d.1 Origen y causa de la falla



```
Se produjo la apertura automática del interruptor 52F3 de S/E Estancilla, correspondiente a la línea 44 kV Estancilla - FFCC Andes N°2, por operación de su protección de distancia residual en zona 1, ante la existencia de un cortocircuito monofásico a tierra, debido al contacto de un ave entre el conductor y el soporte de la estructura N°82 de la mencionada línea.
```

Imagen 4 “Causa descrita en informe de falla”

```
def identificar_categoria(texto):
    texto = texto.lower()
    categoria_detectada = None

    for categoria, frases in categorias.items():
        for frase in frases:
            if frase.lower() in texto:
                categoria_detectada = categoria
                break
        if categoria_detectada:
            break

    if categoria_detectada:
        return categoria_detectada
    else:
        return "Causa no está descrita"
```

Código Python N°6: Proceso para identificar categoría de la falla.

2. En el siguiente ejemplo se está realizando una eliminación de partes innecesarias, ya que el dato que se necesitaba para esta ocasión solo lo esencial era el nombre de la comuna:

h. Comuna donde se presenta la falla

05704: Panquehue.

Imagen 5 “Comuna afectada declara en informe de falla”

```
def limpiar_resultado_comuna(texto):  
    if ":" in texto:  
        texto = texto.split(": ", 1)[1]  
    if texto.endswith("."):  
        texto = texto.rstrip(".")  
    return texto.strip()
```

Código Python N°7: Proceso para limpiar

4.5 Descripción de los Datos Procesados

Para el desarrollo del modelo predictivo, se trabajó con un conjunto de datos que incluye registros históricos de fallas eléctricas recopilados desde el año 2020 hasta el año 2024. En total, se procesaron **100 registros de eventos de falla**, los cuales fueron clasificados según su causa principal. Las categorías de fallas más representativas fueron:

- Condiciones meteorológicas adversas (32%) de las cuales algunas fueron por nieve, temperatura, viento, entre otros.
- Interferencia de vegetación/fauna (20%) de las cuales en su totalidad fueron por impacto de aves.
- Fallas en equipos, componentes eléctricos o de prevención (20%)
- Falla por terceros (10%)
- Otros factores (18%)

Junto con los eventos de falla, se recopilaron también más de **40.000 registros correspondientes a condiciones normales (sin falla)**, los cuales permiten entrenar al modelo con ambas clases: *falla* y *no falla*. Esta distinción es fundamental para construir un modelo de clasificación supervisado que pueda identificar los patrones que preceden a una falla frente a aquellos asociados al funcionamiento normal del sistema.

5 Implementación del Modelo Predictivo

En la siguiente sección llamada Implementación del Modelo Predictivo se detallará el proceso de diseño, desarrollo y entrenamiento de dicho modelo para estimar la probabilidad de fallas en las redes eléctricas por la parte de transmisión. La implementación de este modelo se hizo mediante una **red neuronal densamente conectada (FNN)**, la cual corresponde a la topología más básica dentro de las arquitecturas de redes neuronales artificiales. Esta topología permite establecer una línea base de desempeño y evaluar la capacidad predictiva del modelo en un entorno controlado.

La elección de esta red se debe a su simplicidad, facilidad de entrenamiento y efectividad inicial para tareas de clasificación binaria como la detección de fallas. Sin embargo, se considera que en futuras iteraciones este modelo evolucione hacia una arquitectura **LSTM (Long Short-Term Memory)**, la cual es más adecuada para trabajar con datos secuenciales o series temporales, como los que se presentan en el monitoreo de condiciones meteorológicas y registros históricos de eventos.

5.1 Datos de Entrenamiento

El desempeño del modelo predictivo depende en gran medida de la calidad de los datos utilizados durante su entrenamiento. Para este propósito, se recopilaron y emplearon datos provenientes de registros históricos de fallas eléctricas, complementados con información meteorológica obtenida de herramientas especializadas. A continuación, se detallan con mayor profundidad las características y fuentes de los datos empleados en el desarrollo del modelo.

1. Variables utilizadas

Las variables que han sido seleccionadas para el funcionamiento de este modelo son aquellas cuya relevancia influye en la ocurrencia de una falla eléctrica en términos de condiciones climáticas de las cuales estas se clasificaron en dos categorías las cuales se explicaran a continuación.

i) Variables Meteorológicas

En relación con las variables meteorológicas se recolectaron desde la herramienta de VisualCrossing cómo se detalló en el apartado de extracción de datos y la identificación de fuentes de información se seleccionaron 5 variables de las cuales son las que principalmente afectan a las redes eléctricas en cuanto a las fallas de condiciones climáticas:

- Temperatura (°C).
- Humedad (%).
- Velocidad del Viento (km/s).
- Precipitaciones de la Hora (mm).
- Precipitaciones del Día (mm).

ii) Variables Temporales

En cuanto a las variables temporales al igual que las variables meteorológicas estas se recolectaron desde los informes de fallas y a su vez de la herramienta de VisualCrossing, las variables que han sido seleccionadas son:

- Fecha.
- Hora.

2. Set de datos

Respecto al set de datos que se utilizó fue un archivo .xlsx el cual contiene la información de las fallas registradas desde 2020 hasta 2024 de la V región y las condiciones climáticas recolectadas de todos los días desde 2020 hasta 2024. A continuación, se mostrarán algunas evidencias de el set de datos y el cómo se recolectó.

Se realizó un script vía el lenguaje Python para la recolección de los datos climáticos del transcurso de los tiempos de los años desde el 2020 hasta el 2024:

```
data_list = []
for day in data['days']:

    fecha = day['datetime']
    fecha_formateada = datetime.strptime(fecha, "%Y-%m-%d").strftime("%d/%m/%Y")
    precip_dia = day['precip']
```

```

for hour in day['hours']:
    resultado = {
        "Fecha": fecha_formateada,
        "Hora": hour['datetime'],
        "Temperatura": hour['temp'],
        "Humedad": hour['humidity'],
        "Precipitación": hour['precip'],
        "Velocidad del viento": hour['windspeed'],
        "Precipitacion del dia" : precip_dia
    }
    data_list.append(resultado)

```

Código Python N°8: Proceso para guardar condiciones climáticas.

Fecha	Hora	Temperatura	Humedad	Precipitación por hora	Precipitacion por dia	Velocidad del viento	Dirección del viento	class
01/01/2020	00:00:00	20,2	40,5	0	0	13,8	156	0
01/01/2020	01:00:00	20	40,03	0	0	14,8	150	0
01/01/2020	02:00:00	19	45,63	0	0	11,2	150	0
01/01/2020	03:00:00	17,4	49,23	0	0	11,6	194	0
01/01/2020	04:00:00	17	51,75	0	0	7,6	150	0
01/01/2020	05:00:00	17	48,31	0	0	9,4	150	0
01/01/2020	06:00:00	15,7	57,95	0	0	1,9	98	0
01/01/2020	07:00:00	15	58,79	0	0	1,8	120	0
01/01/2020	08:00:00	16	55,14	0	0	0	0	0
01/01/2020	09:00:00	19,2	45,86	0	0	3,7	180	0
01/01/2020	10:00:00	22	37,92	0	0	9,4	200	0
01/01/2020	11:00:00	24	33,6	0	0	11,2	180	0
01/01/2020	12:00:00	26,1	29,8	0	0	13,1	173	0
01/01/2020	13:00:00	29	26,79	0	0	24,1	160	0
01/01/2020	14:00:00	29	28,66	0	0	25,9	210	0
01/01/2020	15:00:00	28,8	31,72	0	0	25,3	210	0
01/01/2020	16:00:00	29	30,66	0	0	33,5	210	0
01/01/2020	17:00:00	30	28,94	0	0	31,7	200	0

Imagen 6 "Set de datos para Modelo"

5.2 Arquitectura

El modelo predictivo implementado se basa en una red neuronal densamente conectada (FNN) la cual está compuestas de diferentes capas de las cuales estas procesan las entradas para capturar patrones complejos de datos. Dentro de las principales capas utilizadas para este proceso podemos destacar cuatro tipos de las cuales serían una capa de entrada, dos capas ocultas y una capa de salida.

- **Capa de Entrada**

La capa de entrada recibe los datos meteorológicos normalizados. Tiene una forma igual al número de características utilizadas (en este caso, cinco variables de entrada).

- **Capas Ocultas**

El modelo utiliza dos capas ocultas densas totalmente conectadas:

- La primera capa densa consta de 64 neuronas y utiliza la función de activación ReLU.
- La segunda capa densa consta de 32 neuronas, también con activación ReLU.

- **Capa de Salida**

En relación con la capa de salida se utilizó una capa la cual tiene como objetivo generar una predicción que indique la probabilidad de una ocurrencia a falla además de tener en si una función de activación la cual es llamada Sigmoid que permite la salida en un valor entre 0 y 1 indicando como 0 que no es falla y 1 que es falla.

```
model = keras.Sequential([
    keras.layers.Input(shape=(X_train_resampled.shape[1])),
    keras.layers.Dense(64, activation='relu'),
    keras.layers.Dense(32, activation='relu'),
    keras.layers.Dense(1, activation='sigmoid')
])
```

Código Python N°9: estructuración de Capas

- **Uso de la librería Keras**

En la anterior muestra de código se utiliza la librería de keras en python la cual permite construir, entrenar y evaluar redes neuronales de manera sencilla e intuitiva, ofreciendo una sintaxis clara que facilita la experimentación con diferentes arquitecturas y funciones de activación.

5.3 Hiperparámetros

En la siguiente aparatado se explicarán a detalle los hiperparámetros configurados durante el desarrollo y el entrenamiento, para dar un poco de contexto de lo que se trata este aparatado de hiperparámetros son los valores predefinidos que controlan el proceso de aprendizaje y optimización del modelo realizado, en este proceso los hiperparametros se separaron en dos partes la primera de preprocesamiento de datos y la segunda asociada a la configuración de la red neuronal.

1. Preprocesamiento de Datos

Para el preprocesamiento de los datos se ocuparon dos técnicas de undersampling estas técnicas que se ocuparon en este proceso abordan el desbalance de datos, que en sentido de este modelo se basa el desbalance debido a la cantidad de registros de falla desde 2020 hasta 2024 donde se utiliza un registro de 100 fallas eléctricas y el registro de condiciones climáticas desde 2020 hasta 2024 donde se utilizaron un total de más de 40.000 registros a continuación se explicaran las dos técnicas undersampling utilizadas:

- **Random Undersampling (RAS)**

Esta técnica consiste en seleccionar y descartar aleatoriamente las muestras de la clase mayoritaria para lograr un equilibrio en los datos, en este caso se utilizó para que la case mayoritario (no falla) no opaque a la clase minoritaria (falla) y así permita al modelo aprender patrones de forma balanceada, se muestra un extracto de script para la representación de la técnica

```
#estrategia de muestreo
sampling_strategy = {0: majority_class_count // us_factor,
                    1: minority_class_count}
```



```
rus = RandomUnderSampler(sampling_strategy=sampling_strategy, random_state=42)
X_train_resampled, y_train_resampled = rus.fit_resample(X_train, y_train)
```

Código Python N°10: Uso de técnica RAS

- **Edited Nearest Neighbours (ENN)**

El Edited Nearest Neighbours (ENN) es una técnica utilizada para abordar el desbalance de clases, especialmente útil cuando se desea refinar la frontera de decisión entre clases.

De esta manera, ENN realiza un filtrado más enfocado, con el objetivo de aumentar la diferenciación entre clases y limpiar los datos ruidosos en la frontera de decisión, se mostrará un pequeño extracto de script para la representación técnica

```
enn = EditedNearestNeighbours(n_neighbors=enn_neighbors)
X_train_resampled, y_train_resampled =enn.fit_resample(X_train_resampled,
y_train_resampled)
X_train_resampled = np.array(X_train_resampled)
X_test = np.array(X_test)
```

Código Python N°11: Uso de técnica ENN.

2. Configuración de la Red Neuronal

En dicha sección se explicará a detalle la configuración de la red neuronal de este modelo incluyendo así hiperparámetros relacionados con la estructura y el proceso de aprendizaje a continuación se detallará una lista de todas las configuraciones realizadas para este proceso:

- **Número de Neuronas en las Capas Densas**

- i. Primera Capa Densa: Un total de 64 neuronas.
- ii. Segunda Capa Densa: Un total de 32 neuronas.

- **Optimizador**

Para este proceso se utilizó un optimizador llamado **Adaptive Moment Estimation (Adam)**, ya que con uso se logrará ajustar automáticamente la tasa de aprendizaje durante el entrenamiento.

- **Función de Perdida**

La función de perdida que ha sido seleccionada para este proceso fue la de **Binary Cross-Entropy**, ya que es la adecuada para las funciones de clasificación binaria.

- **Función de Activación**

Como es mencionado anteriormente se utilizó una de salida utilizando una función de activación llamada **Sigmoid**, la cual esta función transforma las salidas en valores respectivos entre el 0 y 1 interpretando, así como probabilidades.

- **Función de ReLu**

Además de la función de activación Sigmoid utilizada en la capa de salida, se empleó la función ReLU en las capas intermedias del modelo. ReLU es una función no lineal ampliamente utilizada en redes neuronales debido a su simplicidad computacional y efectividad para resolver problemas de desvanecimiento del gradiente.

3. Entrenamiento del Modelo

El modelo fue entrenado utilizando un conjunto de datos previamente normalizado y balanceado mediante técnicas de submuestreo. El proceso de entrenamiento se configuró con los siguientes parámetros principales:

- **Épocas (epochs):** 100 iteraciones completas sobre el conjunto de entrenamiento. Este número permite que el modelo tenga suficientes oportunidades para aprender patrones.
- **Tamaño del lote (batch size):** 512 el cual es un monto bastante elevado que principalmente se usa para el desbalanceo de clases ya que se cuentan más de 40.000 no fallas y 100 fallas.
- **Pesos de clase ajustados:** Dado el desbalance natural del conjunto de datos (mayoría de registros sin falla), se aplicaron **pesos de clase** para penalizar los errores cometidos sobre la clase minoritaria (fallas eléctricas).

5.4 Validación

La validación del modelo se llevó a cabo utilizando un enfoque de evaluación basado en la partición del conjunto de datos. Para ello, se dividieron los datos disponibles en dos subconjuntos: el 80% de los datos fue utilizado para el entrenamiento del modelo, mientras que el 20% restante se destinó a la fase de prueba o validación.

A continuación, se mostrarán algunas de las métricas para medir el desempeño:

1. Precisión

La precisión del modelo osciló entre el 80% y el 85%, lo cual indica que el modelo clasifica correctamente la mayoría de las observaciones.

2. Recall

Dado que la predicción de fallas (clase minoritaria) es crucial, se priorizó mejorar el recall, alcanzando un valor promedio del 60%. Esto significa que el modelo logra detectar un porcentaje considerable de los eventos de falla reales.

3. Pérdida

En la mayoría de las iteraciones de prueba, la pérdida no caía más allá de los 50%, pero en busca de un mejor recall, con los hiperparametros finales no baja de los 60%

A continuación, se mostrará una imagen donde se ve la validación:

```
263/263 ██████████ 0s
Validación del Modelo
Precisión: 82.03%
Perdida: 67.36%
Recall: 52.00%
```

Imagen 7 “iteración ejemplo”

6 Conclusión

El presente trabajo aborda desde el levantamiento de fuentes de información junto a un proceso de limpieza y transformación hasta la implementación de una herramienta predictiva la cual es un modelo predictivo el cual, a través de su validación se puede estimar una probabilidad de una ocurrencia a falla con base en condiciones y datos temporales históricos, lo que permitiría a las empresas del sector tomar decisiones preventivas y mejorar la gestión de sus recursos operativos

Uno de los principales desafíos encontrados fue la limitada cantidad de datos relacionados con eventos de falla. En total, se trabajó con cerca de 100 registros de fallas, lo que representa un volumen bajo en comparación con los datos sin falla, generando un desbalance en el conjunto. Esta escasez no solo afecta el entrenamiento del modelo, sino que también limita su capacidad de generalización y precisión al aplicarlo en escenarios reales.

A pesar de esta limitación, la experiencia adquirida permitió establecer una metodología clara para el tratamiento de los datos, así como una primera validación del enfoque predictivo. Los resultados obtenidos, si bien aún no concluyentes en términos de precisión, sirven como punto de partida para desarrollos futuros más robustos.

Se recomienda en futuras iteraciones que el modelo predictivo implementado se base en una red neuronal LSTM ya que en la red utilizada es una red densamente conectada (FNN) la cual aspira a ser una LSTM dado que se beneficiaría con respecto a el análisis de series temporal, otra recomendación es el aumento de la cantidad de datos que para este proceso se podría decir que para mejorar la capacidad del modelo se necesitaran una mayor cantidad de datos históricos recolectado.

En resumen, este trabajo demuestra la viabilidad técnica de aplicar modelos de aprendizaje automático a la predicción de fallas en redes eléctricas. Si bien aún hay camino por recorrer, se establece una base sólida que puede ser perfeccionada mediante un enfoque iterativo y el fortalecimiento de los recursos de datos disponibles.

Agradecimientos:

Para empezar este apartado agradecer especialmente a mi familia por su gran apoyo y motivación que me brindan a la hora de iniciar todo este proceso en este ámbito académico universitario, a mis amistades las cuales me dieron un aliento constante y recalando siempre que esta parte académica en la cual estoy presente la puedo llevar a cabo, a compañeros de la carrera que con ellos este proceso académico de los cuales se lograron momentos de aprendizaje inmemorables que enriquecieron esta experiencia y hacer una mención especial a los integrantes del grupo de trabajo llamado “**ElecpreTx**” los cuales fueron apoyo incondicional para este último proceso. Para finalizar me gustaría hacer saber a todos aquellos mencionados en este apartado expresarles mi más sincera gratitud por el apoyo brindado durante el transcurso de este proceso académico universitario.



7 Referencias

- (1) Coordinador Eléctrico Nacional. (n.d.). *Estudios de análisis de falla*. Recuperado de <https://www.coordinador.cl/operacion/documentos/estudios-operacionales/estudios-de-analisis-de-falla/>
- (2) Visual Crossing. (n.d.). *Timeline weather API documentation*. Recuperado de <https://www.visualcrossing.com/resources/documentation/weather-api/timeline-weather-api/>
- (3) WTS Energy. (2023, junio 5). *Red eléctrica*. Recuperado de <https://www.wtsenergy.com/es/glossary/red-electrica/>
- (4) Electric Test. (n.d.). *Factors affecting the safety of transmission lines*. Recuperado de <https://es.electric-test.com/info/factors-affecting-the-safety-of-transmission-l-52987299.html>
- (5) BBVA. (11 de septiembre de 2024). *¿Qué son las redes neuronales? Aplicaciones, tipos y ejemplos*. Recuperado de <https://www.bbva.com/es/innovacion/que-son-las-redes-neuronales-aplicaciones-tipos-y-ejemplos/>
- (6) LinkedIn. (n.d.). *What are the best techniques for undersampling in machine learning?*. Recuperado de <https://www.linkedin.com/advice/1/what-best-techniques-undersampling-machine-learning-2en4c?lang=es&originalSubdomain=es>
- (7) Chioka, S. (2013, Agosto 30). *Class imbalance problem*. Recuperado de <https://www.chioka.in/class-imbalance-problem/>
- (8) Masters in data science. (2022, Abril). *Undersampling*. Recuperado de <https://www.mastersindatascience.org/learning/statistics-data-science/undersampling/>