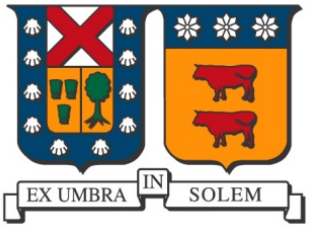
UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
CIUDAD - CHILE



“DISEÑO E IMPLEMENTACIÓN DE ALGORITMO DE MACHINE LEARNING CAPAZ DE PREDECIR Y DETECTAR PATRONES ANÓMALOS EN LAS SERIES DE TIEMPO ASOCIADAS A LA GENERACIÓN E INGESTA DE EVENTOS DE TELEMETRÍA CERCANO A TIEMPO REAL PARA GAUSS CONTROL”

PEDRO ISAAC HERRERA SOTO

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA

Profesor Guía: Dr. Juan Ricardo Ñanculef Alegría
Profesor Correferente: Sr.Darío Muñoz

Agosto - 2024

DEDICATORIA

A mi amada y extraordinaria familia, mis entrañables amigos y todos aquellos que me acompañaron en este largo proceso: su apoyo incondicional ha sido el pilar de mi éxito. Sin ustedes, nada de esto habría sido posible. Mi corazón rebosa de gratitud por cada gesto, palabra de aliento y momento compartido. Agradezco profundamente a Dios por guiar mis pasos y premiarme con personas tan maravillosas en mi vida. Su presencia ha sido mi fortaleza durante este desafío.

AGRADECIMIENTOS

Quiero expresar mi más sincero agradecimiento a las siguientes personas, cuyo apoyo y orientación fueron fundamentales para la realización de esta tesis: Al Sr. Darío Muñoz, jefe del Área de Ciencia de Datos de Gauss Control, por su gran ayuda en la resolución de los diversos desafíos que surgieron durante el desarrollo de esta memoria. Su confianza en mi capacidad para llevar a cabo este proyecto fue una motivación constante. Al Sr. Andrés Gibson, Gerente de Tecnología de Gauss Control, por su valiosa contribución en la definición de la arquitectura del modelo y la selección de la tecnología más apropiada para este trabajo, Y Ángelo Miranda, por su constante apoyo durante la fase de programación del modelo y por compartir sus ideas innovadoras, las cuales enriquecieron significativamente este proyecto.

RESUMEN

Resumen— Esta memoria aborda el problema que enfrenta la empresa Gauss Control al detectar anomalías en las series de tiempo de ingesta de eventos de telemetría en tiempo real. Se propone una solución basada en dos modelos de *Variational Autoencoder (VAE)*, uno especializado en patrones de días laborales y otro en patrones de fines de semana/días festivos. En el Capítulo 1 se define el problema, los procesos actuales de detección de anomalías y sus limitaciones. El Capítulo 2 presenta el marco conceptual con conceptos de inteligencia artificial aplicados a series de tiempo. En el Capítulo 3 se detalla la propuesta de solución, incluyendo la arquitectura del sistema con los dos modelos *VAE*, su entrenamiento, detección de anomalías y evaluación de rendimiento. El Capítulo 4 valida la solución usando casos reales de anomalías proporcionados por clientes, comparándola con otros métodos. Finalmente, el Capítulo 5 presenta las conclusiones, discutiendo alcances, limitaciones, contribuciones, aplicaciones y recomendaciones futuras. Los resultados muestran que el enfoque propuesto basado en *VAE* es prometedor y supera a otros métodos en la detección de anomalías en series de tiempo. Permite a Gauss Control anticiparse y responder rápidamente a eventos anómalos, mejorando la calidad del servicio y satisfacción del cliente.

Palabras Clave— Palabras Clave: Detección; anomalías; series; tiempo; VAE

ABSTRACT

***Abstract*—** This memory addresses the problem faced by Gauss Control when detecting anomalies in real-time telemetry event ingestion time series. A solution based on two Variational Autoencoder (VAE) models is proposed, one specialized in weekday patterns and the other in weekend/holiday patterns. Chapter 1 defines the problem, the current anomaly detection processes, and their limitations. Chapter 2 presents the conceptual framework with concepts of artificial intelligence applied to time series. Chapter 3 details the proposed solution, including the system architecture with the two VAE models, their training, anomaly detection, and performance evaluation. Chapter 4 validates the solution using real anomaly cases provided by customers, comparing it with other methods. Finally, Chapter 5 presents the conclusions, discussing scope, limitations, contributions, applications, and future recommendations. The results show that the proposed VAE-based approach is promising and outperforms other methods in detecting anomalies in time series. It allows Gauss Control to anticipate and respond quickly to anomalous events, improving service quality

and customer satisfaction.

Keywords— Detection; anomalies; series; time; VAE

GLOSARIO

UTFSM: Universidad Técnica Federico Santa María.
LSTM: *Long Short-Term Memory*, Es un tipo de red recurrente donde la información persiste.
ENCODER-DECODER: Es una arquitectura de red neuronal utilizada para tareas de secuencia a secuencia, donde el codificador procesa la entrada y el decodificador genera una salida basada en la representación codificada.
VAE: Modelo generativo en el que se combina un *autoencoder* con técnicas de inferencia *variacional*.
GAN: Modelo generativo en el que se combina un generador con un discriminador.
Denegación de servicio: En esta memoria una Denegación de Servicio se refiere a cualquier incidente que impida a los servicios recibir datos. Esto puede ser causado por diversos factores, desde un ataque informático hasta un fallo técnico como una fuga de memoria (*memory leak*).
Evento: En esta memoria un evento es un tipo de dato que posee Gauss control.

ÍNDICE DE CONTENIDOS

RESUMEN iv

ABSTRACT iv

GLOSARIO vi

ÍNDICE DE FIGURAS ix

ÍNDICE DE TABLAS ix

INTRODUCCIÓN 1

CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA 2
1.1 Contexto . . . 2
1.2 SITUACIÓN ACTUAL . . . 3
1.3 PROCESOS . . . 3
1.3.1 PROCESO DE IDENTIFICACIÓN DE ANOMALÍA . . . 3
1.3.2 Creación de *ticket* . . . 5
1.3.3 Indagación y solución de la anomalía . . . 7
1.4 PROBLEMAS DETECTADOS . . . 8
1.5 OBJETIVOS . . . 9
1.5.1 Objetivo General . . . 9
1.5.2 Objetivos específicos . . . 9

CAPÍTULO 2: MARCO CONCEPTUAL 10
2.1 Inteligencia artificial y automatización . . . 10
2.1.1 Ventajas . . . 10
2.1.2 Desventajas . . . 10
2.2 MODELOS UTILIZADOS . . . 11
2.3 Indicadores de Gestión . . . 12
2.4 Desarrollo . . . 12
2.4.1 Stack Tecnológico . . . 12
2.5 Arquitectura del modelo vae . . . 14
2.5.1 Encoder LSTM . . . 15
2.5.2 Capa de muestreo VAE . . . 16
2.5.3 Decoder LSTM . . . 17
2.5.4 Función de perdida . . . 17
2.5.5 Proceso de detección de anomalías . . . 18
2.5.6 Entrenamiento . . . 18

CAPÍTULO 3: PROPUESTA DE SOLUCIÓN 20
3.0.7 Arquitectura del modelo vae . . . 20
3.1 Arquitectura del Sistema Propuesto . . . 22

3.1.1 Cronjob . . . 24
3.1.2 Función que verifica la fecha del último entrenamiento . . . 24
3.1.3 Función de entrenamiento . . . 25
3.1.4 Entrenamiento de los Modelos VAE . . . 25
3.1.5 Preprocesamiento de Datos . . . 25
3.1.6 VAE para días de semana . . . 25
3.1.7 VAE para Fines de Semana y Días Festivos . . . 25
3.1.8 Ajuste de Hiperparámetros . . . 26
3.2 función de evaluación . . . 26
3.2.1 Vae Días de semana . . . 26
3.2.2 Vae Fin de semana . . . 26
3.2.3 Vae total . . . 27
3.2.4 Umbrales de Anomalía . . . 27
3.2.5 Notificación de Anomalías . . . 27
3.3 Evaluación del Rendimiento . . . 29
3.4 Evaluación del Rendimiento . . . 29
3.4.1 Costo oportunidad vs Costo de implementación . . . 29
3.5 Hiper parámetros propuestos . . . 30

CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN **31**
4.1 Validación de la Solución con Anomalías Reales . . . 31
4.1.1 Caso de Anomalía 1 . . . 31
4.1.2 Caso de Anomalía 2 . . . 32
4.1.3 Caso de Anomalía 3 . . . 33
4.1.4 Resultados . . . 34
4.2 Validación de la Solución con *dataset* de anomalías . . . 34
4.2.1 Resultados . . . 35
4.3 Conclusiones Preliminares . . . 36

CAPÍTULO 5: CONCLUSIONES **37**
5.1 Conclusiones Generales . . . 37
5.2 Cumplimiento de objetivos . . . 38
5.3 Trabajo futuro. . . . 39
5.3.1 Ajuste Automático de Umbrales. . . . 39
5.3.2 Enfocarlo en tipos de eventos específicos. . . . 39
5.3.3 Ajuste de hiperparametros acorde al cliente. . . . 40
5.3.4 Visualización de la anomalía. . . . 40
5.3.5 Entrenamiento personalizado. . . . 40
5.4 Palabras finales del autor. . . . 41

REFERENCIAS BIBLIOGRÁFICAS **42**

ÍNDICE DE FIGURAS

1 Evolución puntos *GPS* por cliente. Fuente: Gauss Control 4
2 Diagrama de flujo detección de anomalía. Fuente: Elaboración propia 5
3 *ticket* 6
4 Diagrama de flujo creación de *ticket*. 7
5 Diagrama de indagación y solución anomalía. Fuente: Elaboración propia . . 8
6 Arquitectura LSTM Encoder del modelo VAE 15
7 Arquitectura Decoder del modelo VAE 17
8 Arquitectura detección de anomalías 19
9 Arquitectura general del modelo vae 21
10 Diagrama de Arquitectura del modelo 23
11 Diagrama del modelo VAE 28
12 Gráfico comparativo de ambos modelos frente a una anomalía. 31
13 Gráfico comparativo de ambos modelos frente a una anomalía. 32
14 Anomalía detectada por el modelo. 33

ÍNDICE DE TABLAS

1 Comparación de rendimiento VAE v/s Prophet. 34
2 Comparación de rendimiento *Vae* para un *dataset* diferente al de Gauss.(Parte 1). 34
3 Comparación de rendimiento *Vae* para un *dataset* diferente al de Gauss. (Parte 2). 35

INTRODUCCIÓN

En la era digital actual, la gestión eficiente de grandes volúmenes de datos de telemetría en tiempo real es crucial para muchas empresas. Estos datos son fundamentales para monitorear el rendimiento de productos y servicios, gestionar riesgos y prevenir incidentes. Sin embargo, el creciente volumen y complejidad de estos datos plantea un desafío significativo para la detección oportuna de anomalías.

Gauss Control, una empresa innovadora en predicción y prevención de accidentes, enfrenta este reto en su operación diaria. Su plataforma avanzada integra datos de múltiples fuentes y utiliza inteligencia artificial para predecir y gestionar riesgos. No obstante, la empresa reconoce que las interrupciones en el flujo de datos o la pérdida de información pueden tener consecuencias graves, como la incapacidad de monitorear adecuadamente los sistemas de los clientes, aumentando el riesgo de accidentes.

Actualmente, Gauss Control depende del monitoreo manual para identificar anomalías, un método propenso a errores y difícil de escalar. A medida que la empresa crece y diversifica sus fuentes de datos, se hace evidente la necesidad de una solución más robusta y automatizada para garantizar la integridad y continuidad de sus servicios críticos de prevención de accidentes.

Esta memoria de tesis aborda el desafío al proponer el diseño e implementación de un algoritmo de aprendizaje automático capaz de predecir y detectar patrones anómalos en las series de tiempo asociadas con la generación e ingesta de eventos de telemetría en tiempo real. El enfoque propuesto aprovecha técnicas avanzadas de aprendizaje automático, como los modelos de Variational Autoencoder (VAE), para capturar las características únicas de los patrones de datos tanto en los días laborales como en los fines de semana y días festivos.

A lo largo de esta memoria, se explorará el contexto y los desafíos actuales que enfrenta Gauss Control, se presentará el marco conceptual y los modelos utilizados, se detallará la propuesta de solución y su validación, y se discutirán las conclusiones y recomendaciones derivadas de este trabajo. El objetivo principal es desarrollar una solución escalable y eficiente que permita a Gauss Control anticiparse y responder de manera oportuna a las anomalías, minimizando los riesgos y mejorando la calidad del servicio para sus clientes.

CAPÍTULO 1
DEFINICIÓN DEL PROBLEMA

1.1. Contexto

Las denegaciones de servicio son eventos impredecibles que representan un desafío para las operaciones empresariales. Aunque una empresa pueda contar con una capacidad de respuesta adecuada, los efectos de una denegación de servicio pueden ser desastrosos. En el caso de Gauss Control, esto implica que los usuarios no podrán acceder a su sitio web durante un tiempo determinado, lo que resulta en la pérdida de información vital sobre sus productos y pone en riesgo la capacidad de la empresa para gestionar el riesgo y prevenir accidentes. Además, existe la posibilidad de perder a un cliente valioso para Gauss en caso de que ocurra un accidente.

El problema radica en que las denegaciones de servicio son difíciles de predecir. Aunque se pueda observar una disminución en la cantidad de datos recibidos en un periodo de tiempo. esto implicaría tener que monitorear constantemente el comportamiento de la ingesta de datos durante las 24 horas del día. Sin embargo, esta tarea no es viable cuando existen múltiples fuentes de datos para cada cliente y múltiples clientes en general. Es por esta razón que surge la necesidad de un modelo capaz de predecir las “anomalías” en la ingesta de datos.

La consideración de un modelo de aprendizaje automático para mejorar el proceso de detección y respuesta a anomalías conlleva una evaluación exhaustiva de los aspectos positivos y negativos asociados. Entre los beneficios destacados se encuentra la posibilidad de ganar tiempo de reacción al anticipar eventos anómalos, lo que permite una respuesta más rápida y eficiente. Además, el modelo puede contribuir a evitar denegaciones de servicio y detectar comportamientos erráticos o malas configuraciones en los sensores ya que estos problemas producirían anomalías que el algoritmo detectaría. Sin embargo, es necesario tener en cuenta los perjuicios potenciales, como el costo de implementar la solución en producción y la existencia de falsos positivos. La implementación de un modelo de aprendizaje automático implica recursos técnicos y un proceso riguroso de desarrollo y pruebas, lo que puede generar un costo significativo. Asimismo, los modelos no son perfectos y pueden generar falsas alarmas, lo que requiere esfuerzos adicionales para ajustar y mejorar el modelo.

En conclusión, la evaluación equilibrada de los beneficios y perjuicios asociados con la adopción de un modelo de aprendizaje automático. le ofrecerá una ayuda al equipo humano responsable de la operación entregándole información oportuna sobre la calidad

del servicio y la tendencia que podría prevenir alguna caída o disminución en la calidad y cantidad de datos, que redunda en una baja en la calidad de servicio.

1.2. SITUACIÓN ACTUAL

El proceso general de reacción ante una identificación de anomalía, esta a su vez compuesto por varios subprocesos:

- Identificación de anomalía: Si la ingesta de eventos baja de cierto porcentaje respecto al mismo día, pero de la semana anterior se detectará como una anomalía.
- Creación de *ticket*: Se crea un *ticket* por parte de monitoreo donde se informa a la empresa de telemetría y al jefe de proyecto dicho problema
- Indagación: Se identifica si el problema está presente en los demás clientes y se busca que está ocasionando dicha falla.

Para los procesos recién identificados se identifican los siguientes actores

- Encargado de monitoreo: Encargado de identificar la anomalía además de la creación de un *ticket* notificando el problema
- Jefe de proyecto: Es el encargado de notificar al cliente en el caso de que el problema llegue a mayores

1.3. PROCESOS

1.3.1. PROCESO DE IDENTIFICACIÓN DE ANOMALÍA

Hasta ahora, el proceso para identificación de anomalías comienza con un monitoreo rutinario en la ingesta de eventos donde si se observan comportamientos anómalos se comienza a revisar la evolución en la ingesta de datos para este evento, si la cantidad de datos que se recibieron disminuye o aumenta un 70 % respecto a la media de la data en los últimos 14 días se identificara como anomalía, luego se comparará con la cantidad de data recibida para el mismo día pero de la semana anterior y si el comportamiento es muy diferente se detectará como anomalía y se comenzará el siguiente proceso.

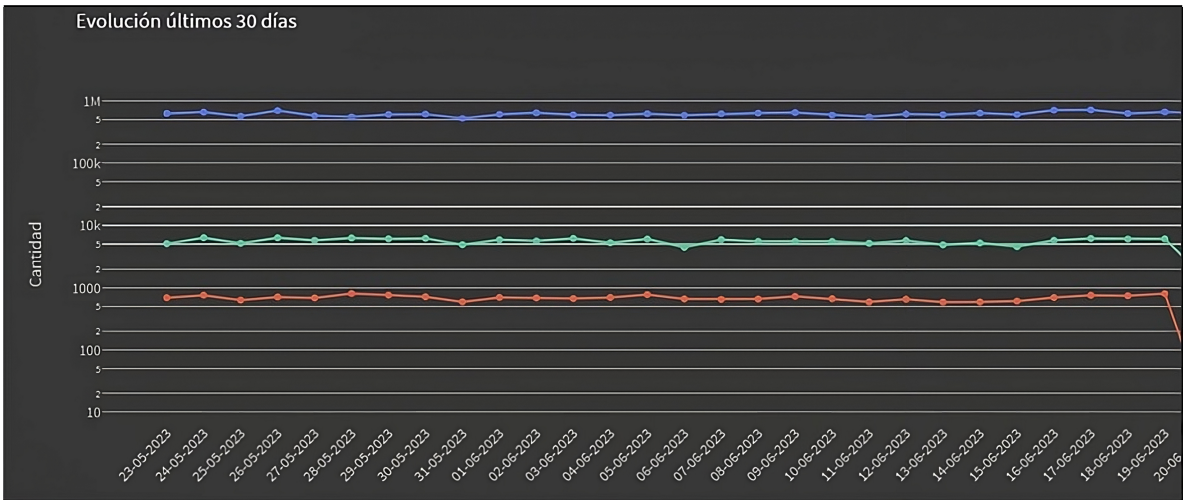


Figura 1: Evolución puntos *GPS* por cliente. Fuente: Gauss Control

Cabe destacar que existen excepciones donde se descubre una anomalía por medio de un reclamo directo del cliente, en estos casos se pasa directamente a la creación de *ticket* y la indagación de la anomalía

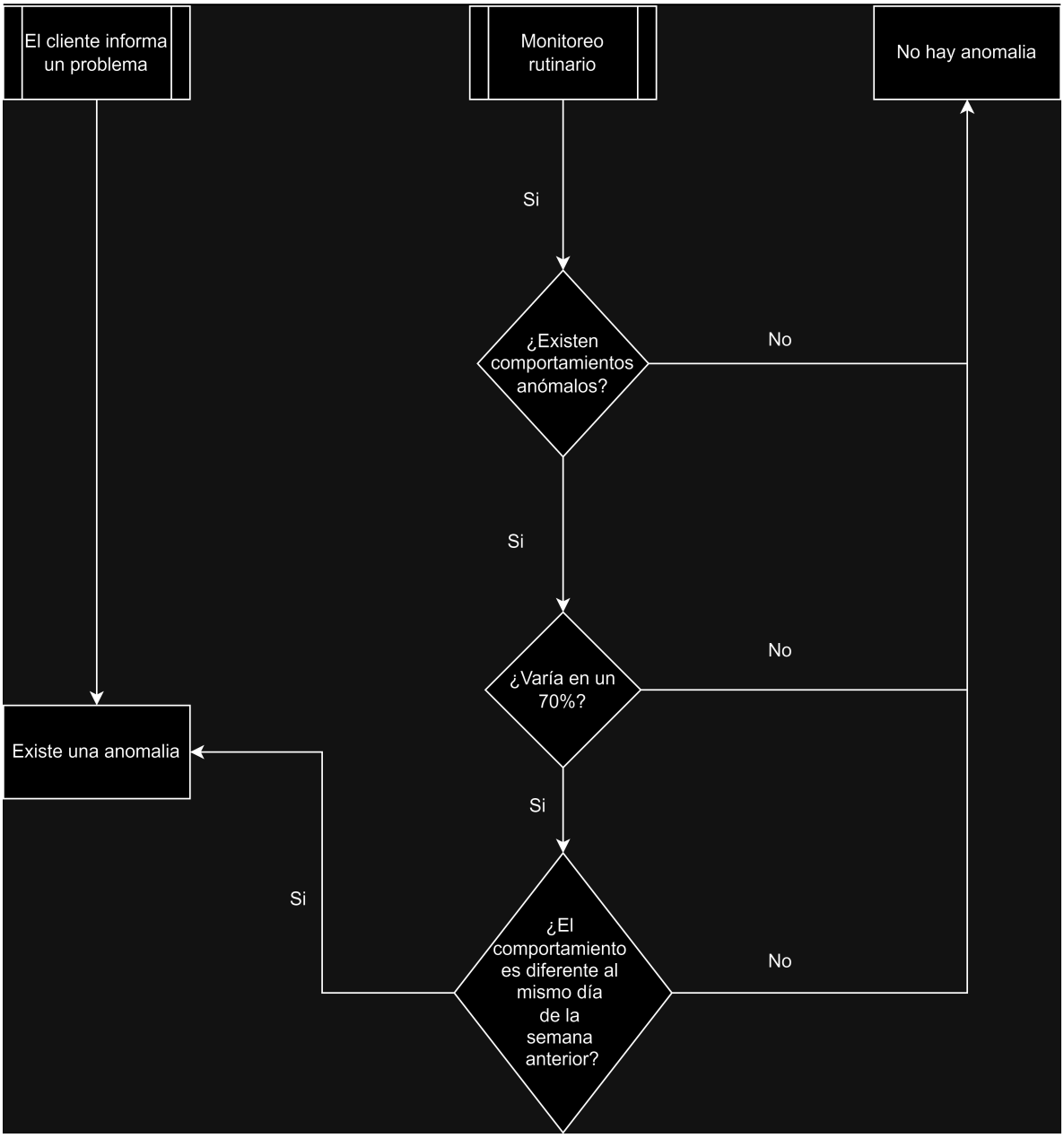


Figura 2: Diagrama de flujo detección de anomalía. Fuente: Elaboración propia

1.3.2. Creación de *ticket*

Luego de identificar una anomalía se notifica al jefe de proyecto junto a todos los involucrados en dicho proyecto. Luego se le envía un correo a la empresa de telemetría donde se les solicita que informen si al momento de inyectar eventos les aparece un error, en caso de ser así se les pide que adjunten dicho error junto a las credenciales y el punto de acceso usado. En el caso de existir una denegación de servicio se notifica a ingeniería para solucionar el

problema.

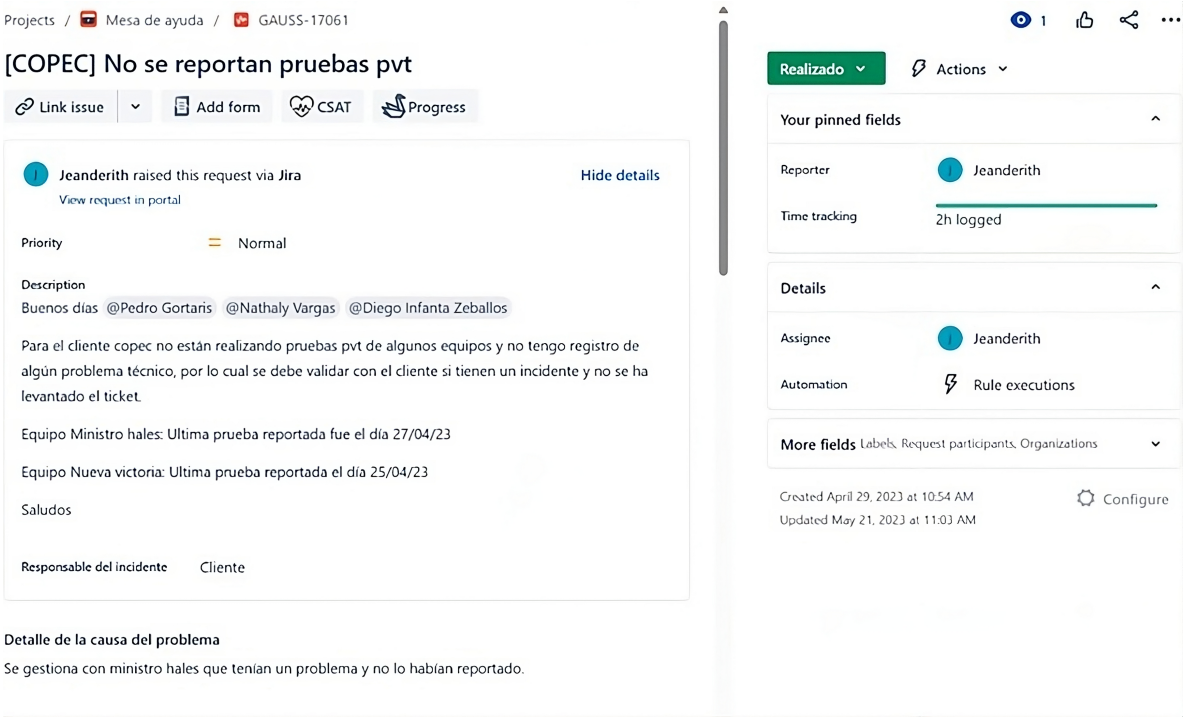


Figura 3: *ticket*
Fuente: Gauss Control

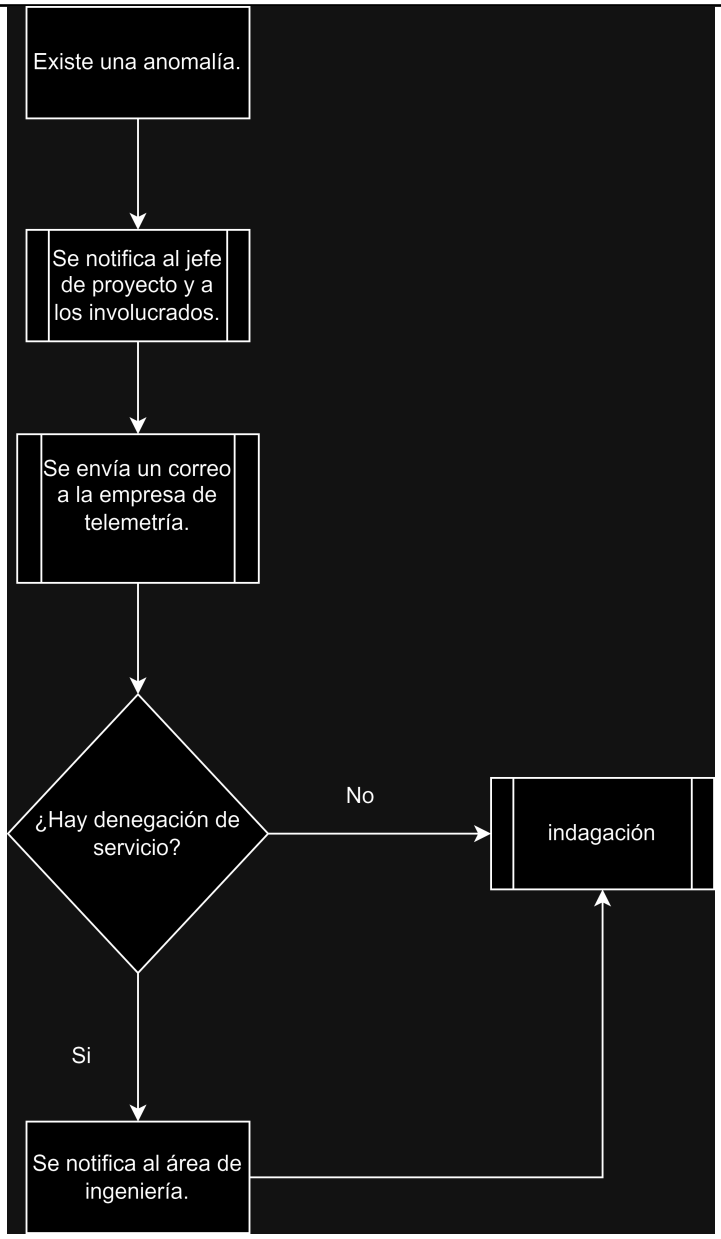


Figura 4: Diagrama de flujo creación de *ticket*.
Fuente: Elaboración propia

1.3.3. Indagación y solución de la anomalía

Luego de que las demás áreas se enteren del problema se comienza la indagación para descubrir que lo puede estar ocasionando.

- En el caso que la anomalía ocurre en gauss, el área de ingeniería comenzará a averiguar

el suceso que ocasionó la denegación para asegurarse que no vuelva a ocurrir

- En el caso de que sea un *bug* esta toma un nivel de prioridad de acuerdo con la gravedad de este y la importancia del cliente.
- En el caso de que la anomalía ocurra en el cliente o la empresa de telemetría contratada por el cliente, los encargados de monitoreo se comunican con el responsable de la ingesta de datos para notificarle que tiene un problema y se le solicita que envíe la información del error que tiene al momento de inyectar datos.

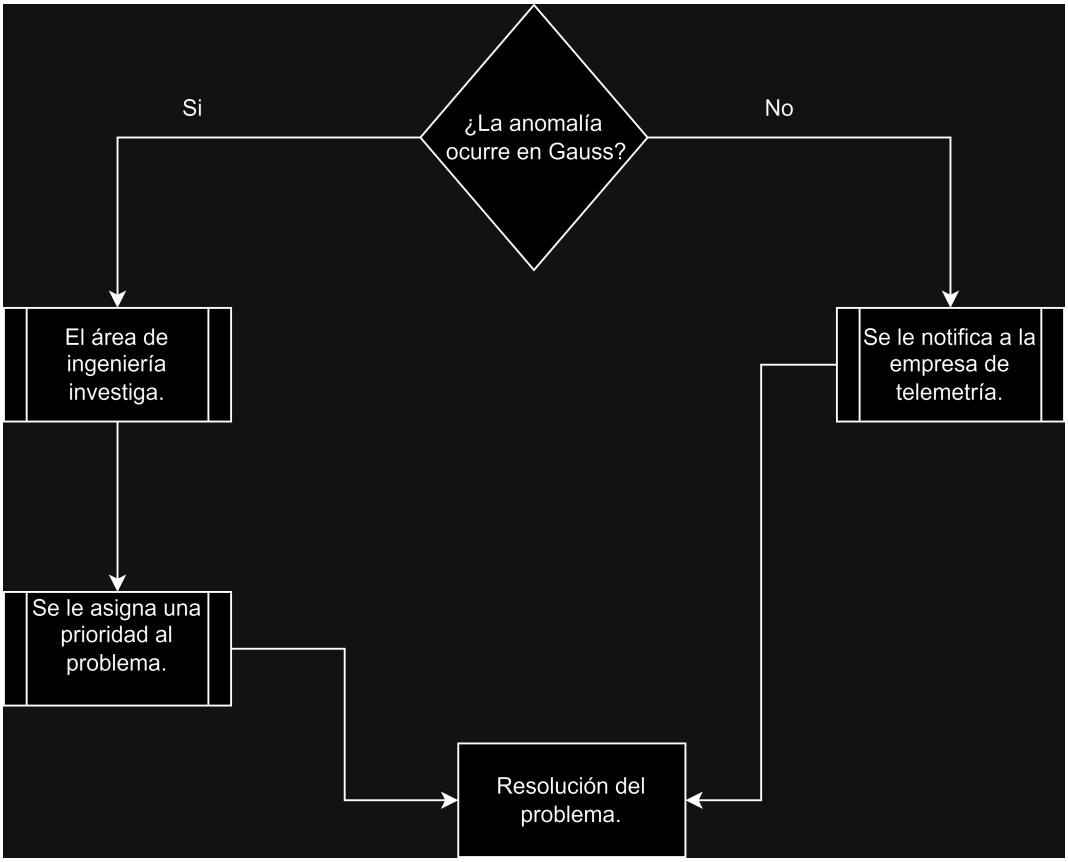


Figura 5: Diagrama de indagación y solución anomalía. Fuente: Elaboración propia

1.4. PROBLEMAS DETECTADOS

En los procesos recién descritos es fácil identificar problemas asociados al momento en que identifican una anomalía.

- Una anomalía solo es identificada luego de que se perdió una gran cantidad de información (alrededor del 70 %).

- El método de monitoreo manual es poco escalable en término de descubrir nuevos clientes ya que se hace imposible monitorear 24 horas al día múltiples eventos que existen a su vez en múltiples clientes.
- Una denegación de servicio solo es detectada una vez que ya sucedió.
- Debido a la detección tardía de una anomalía o denegación de servicio la respuesta debe ser rápida lo que a veces conlleva a no tomar las mejores decisiones.
- Existe la posibilidad que un cliente se dé cuenta del problema antes que la empresa.

1.5. OBJETIVOS

1.5.1. Objetivo General

- Diseñar e implementar un algoritmo de aprendizaje automático capaz de detectar patrones anómalos en la ingesta y/o generación de eventos de telemetría para así resolver la problemática planteada por gauss control.

1.5.2. Objetivos específicos

- Identificar los aspectos relevantes del problema que afectan la detección de patrones y considerarlos en el diseño del algoritmo.
- Comparar y evaluar los resultados del algoritmo en la vida real con los resultados del Estado del Arte.
- Anticiparse y detectar en tiempo y forma anomalías en la ingesta y generación de eventos de telemetría de los productos de Gauss Control.
- Generar una solución a la problemática descrita que considere un balance entre el costo de oportunidad de contar con esta información versus el costo de infraestructura de tener esta solución.
- Trabajar bajo el peor escenario posible de funcionamiento con una granularidad de 24 horas.
- Hacer que la solución sea escalable y auto gestionada en términos de descubrir nuevos clientes como capaz de ser re-entrenado cada cierto tiempo.

CAPÍTULO 2
MARCO CONCEPTUAL

2.1. Inteligencia artificial y automatización

El aumento exponencial de la infraestructura en la nube ha transformado la manera en que las organizaciones almacenan, procesan y acceden a sus datos. A medida que más empresas trasladan sus operaciones y servicios a la nube, la cantidad y complejidad de los datos generados aumenta considerablemente [Raghavendran *et al.*, 2016].

Este aumento explosivo en la generación de datos plantea desafíos únicos, especialmente en lo que respecta a la detección temprana de comportamientos inusuales. A medida que los sistemas en la nube se vuelven más interconectados y las infraestructuras se expanden, también aumentan los riesgos asociados con la seguridad y el rendimiento. La gran cantidad de datos generados dificulta la detección de patrones anómalos. Los enfoques tradicionales de monitoreo y detección de anomalías pueden no ser suficientes para abordar esta complejidad [Vallis *et al.*, 2014].

Es por esto que se comienza a incursionar en la utilización de modelos de aprendizaje automático para la resolución de esta problemática. Sin embargo, como toda solución, esta tiene ventajas y desventajas, las cuales detallaremos a continuación [Choi, 2021].

2.1.1. Ventajas

- **Detección temprana de anomalías:** Los modelos de Machine Learning pueden predecir anomalías en las series temporales, permitiendo tomar medidas antes de que los problemas se vuelvan más costosos.
- **Precisión:** Los modelos tienen la capacidad de analizar grandes volúmenes de datos y descubrir patrones complejos que pueden identificar anomalías, logrando una detección más precisa y reduciendo los falsos positivos.
- **Escalabilidad:** Al manejar grandes volúmenes de datos, los modelos son adecuados para entornos empresariales donde hay numerosas series de tiempo para supervisar.

2.1.2. Desventajas

- **Requiere conjuntos de datos grandes:** Necesitan grandes volúmenes de datos para entrenarse de manera efectiva.

- **Dificultad en la interpretación de resultados:** Algunos modelos de aprendizaje se comportan como cajas negras, lo que significa que sus decisiones pueden ser difíciles de interpretar.
- **Requiere mantenimiento y actualización continua:** Los modelos de aprendizaje automático necesitan ser actualizados y mantenidos a medida que se recopilan nuevos datos.

2.2. MODELOS UTILIZADOS

En 2016, Malhotra sugiere el empleo de un modelo fundado en *Encoder-Decoder* junto con una red LSTM para afrontar este desafío. Esta metodología combina dos elementos esenciales: un codificador (*encoder*) y un decodificador (*decoder*) [Malhotra *et al.*, 2016].

El objetivo del codificador es capturar la información importante del contexto de entrada y comprimirla en una representación vectorial de menor dimensión, conocida como estado oculto. En este caso, la red LSTM se utiliza como el codificador, lo que posibilita la captura de dependencias a largo plazo en los datos de entrada.

Una vez que el codificador ha capturado la información del contexto de entrada, el decodificador utiliza este estado oculto como punto de partida para generar una secuencia de salida. El decodificador también se basa en una red LSTM, que recibe el estado oculto del codificador y genera una secuencia de salida paso a paso.

El modelo emplea una arquitectura de atención, que permite al decodificador enfocarse en partes específicas del estado oculto del codificador mientras genera la secuencia de salida. Los resultados de este modelo fueron esperanzadores ya que consiguieron entre 83 % y 89 % de precisión dependiendo del dataset utilizado.

Si bien el modelo antes mencionado era bastante preciso, aún tenía problemas cuando los conjuntos de datos utilizados estaban desequilibrados. Es por este problema que en 2019 Wenqian Jiang propone la utilización del modelo de 2016 [Malhotra *et al.*, 2016] en conjunto con una red Generative Adversarial Network (GAN) [Jiang *et al.*, 2019]. La GAN está compuesta por un generador y un discriminador, donde el generador se entrena para crear múltiples series de tiempo normales, mientras que el discriminador debe ser capaz de diferenciar entre una serie de tiempo creada por el generador y una real.

Al finalizar el entrenamiento, se utiliza la GAN para generar miles de series de tiempo que serán utilizadas como instancias normales en un modelo de detección de anomalías. La utilización de GAN logra eliminar la problemática que tenía el modelo original, además de aumentar la precisión en la detección de anomalías.

Tres años después, en 2022, Li utiliza el modelo de Jiang como base y cambia la GAN por un Variational Autoencoder (VAE), logrando que el modelo sea mucho más estable y se enfoque

en detectar anomalías en series de tiempo que involucran múltiples variables [Li *et al.*, 2022].

El VAE es un tipo de modelo generativo que aprende una distribución latente subyacente de los datos, permitiendo la generación de nuevas muestras. Este se entrena utilizando únicamente datos normales para capturar el comportamiento normal del sistema.

Una vez que el modelo de detección de anomalías está entrenado, se puede utilizar para detectar anomalías en tiempo real en nuevos datos. El modelo compara la representación latente de los datos de entrada con la distribución de datos normales aprendida durante el entrenamiento y genera una puntuación de anomalía para cada muestra. Las muestras con puntuaciones de anomalía superiores a un umbral predefinido se consideran anormales y se generan alertas.

2.3. Indicadores de Gestión

Para medir las mejoras en el proceso de detección de anomalías en la ingesta de datos en Gauss Control, son necesarios los indicadores de gestión (o KPI por sus siglas en inglés):

- **Tasa de detección de anomalías:** Corresponde al porcentaje de anomalías detectadas correctamente en relación con el número total de anomalías presentes en los datos.
- **Tasa de falsos positivos:** Representa el porcentaje de puntos de datos normales clasificados erróneamente como anomalías.
- **Tiempo de detección:** Es el tiempo promedio que tarda el modelo en detectar una anomalía desde el momento en que ocurre.
- **Precisión:** Es el porcentaje de anomalías detectadas correctamente en relación con el número total de puntos de datos clasificados como anomalías.

2.4. Desarrollo

A continuación, se detallan las herramientas de trabajo que se aplicaran en el desarrollo del proyecto y los motivos por los que fueron seleccionados por sobre las alternativas existentes.

2.4.1. Stack Tecnológico

Cuando se menciona un stack para desarrollo se está hablando de un conjunto de tecnologías que se aplican en conjunto para resolver distintas áreas de un mismo

proyecto. El stack elegido propone el uso de tecnologías conocidas y confiables para desarrollar rápidamente un modelo de aprendizaje automático:

- **Lenguaje de programación:**
  - **Python:** Se utilizó debido a que es una excelente opción para la creación de modelos, dado su amplia adopción en el campo del aprendizaje automático. Python cuenta con una gran cantidad de bibliotecas y marcos de trabajo específicos para el aprendizaje automático.
  - **Node-RED:** Se utilizó debido a que es una herramienta de desarrollo basada en flujos que permite la creación de aplicaciones de Internet de las Cosas (IoT) de manera eficiente y visual.
- **Librerías y marcos de trabajo de aprendizaje automático:**
  - **TensorFlow:** TensorFlow es una biblioteca líder en el campo del aprendizaje automático y la inteligencia artificial.
  - **Scikit-Learn:** Es una biblioteca muy versátil que proporciona una amplia gama de algoritmos de aprendizaje automático listos para usar.
- **Herramientas para procesamiento de datos y visualización:**
  - **Pandas:** Para el procesamiento de datos, Pandas es una excelente opción. Proporciona estructuras de datos flexibles y eficientes.
- **Herramientas de desarrollo e integración:**
  - **Jupyter Notebook:** Es una herramienta muy útil para el desarrollo interactivo y la documentación de modelos de aprendizaje automático.
  - **Git:** Es esencial para el control de versiones y la gestión del código fuente. Al utilizar Git, puedes realizar un seguimiento de los cambios en tu proyecto, colaborar con otros desarrolladores y revertir modificaciones si es necesario.
- **Herramientas de almacenamiento:**
  - **SQLite:** Base de datos relacional liviana, sin servidor, ideal para aplicaciones embebidas o de bajo volumen. Es útil por su simplicidad, portabilidad y bajo consumo de recursos.
  - **BigQuery:** Servicio de almacenamiento y análisis de datos a gran escala de Google Cloud. Es preferible para análisis de grandes volúmenes de datos con alta velocidad y capacidad de procesamiento.
  - **Cloud Storage:** Servicio de almacenamiento de objetos de Google Cloud. Es óptimo para almacenar y acceder a grandes cantidades de datos no estructurados de forma escalable y duradera.

- **Herramientas de inteligencia artificial:**
  - **ChatGPT:** Se utilizó para buscar definiciones de ciertos tecnicismos.
  - **Bing Chat:** Se utilizó para buscar artículos asociados a la problemática.
  - **Claude.ai:** Se empleó para optimizar y refinar la redacción de diversas explicaciones.

2.5. Arquitectura del modelo vae

Como podemos observar la vae está compuesta por: Encoder LSTM: la cual es una red neuronal recurrente diseñada para procesar secuencias de datos y capturar patrones temporales a largo plazo. Esta transforma una secuencia de entrada en un espacio latente de menor dimensión. Este espacio latente representa las características más relevantes de la secuencia original.

Sampling Layer: es una capa que implementa el proceso de reparametrización para generar muestras desde un espacio latente. permite la generación de nuevas muestras latentes que siguen una distribución específica (normalmente una distribución gaussiana). Esto es crucial para el proceso de generación y para la interpretación de las variaciones en los datos.

Decoder LSTM: una red LSTM que toma las muestras del espacio latente y las transforma de nuevo en una secuencia de datos. El decoder LSTM reconstruye la secuencia original (o una versión aproximada de ella) a partir de la representación latente generada por el encoder y la capa de muestreo. Esto permite evaluar la calidad de la codificación y la capacidad de generación del modelo.

2.5.1. Encoder LSTM

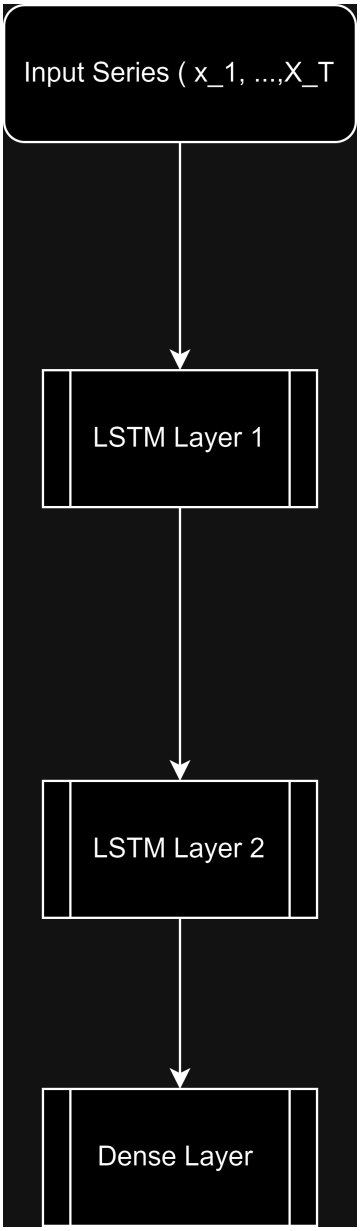


Figura 6: Arquitectura LSTM Encoder del modelo VAE
Fuente: elaboración propia

El encoder LSTM consta de dos capas LSTM, con 64 y 128 neuronas, respectivamente. Además, incluye una capa densa con 96 neuronas que se conecta al espacio latente, el cual tiene una dimensionalidad de 48.

2.5.2. Capa de muestreo VAE

La capa de muestreo tiene una dimensión de 48, por lo que cada capa densa tiene 48 neuronas. Además, esta capa utiliza el truco de reparametrización dado por:

$$z = \mu + \epsilon \cdot \sigma$$

donde $\epsilon \sim (0, 1)$.

2.5.3. Decoder LSTM

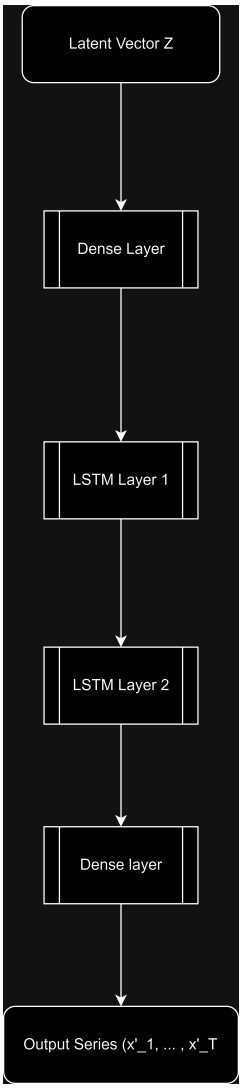


Figura 7: Arquitectura Decoder del modelo VAE
Fuente: Elaboración propia

El *Decoder* LSTM consta de dos capas LSTM, con 128 y 64 neuronas, respectivamente. Además, incluye dos capas densas con 96 y 1 neuronas respectivamente

2.5.4. Función de perdida

La función de pérdida está dada por:

$$L = \text{MSE}(x, x') + \beta \cdot \text{KL}(q(z \mid x) \parallel p(z))$$

donde:

- $\text{MSE}(x, x')$ es el error cuadrático medio entre la entrada original x y la reconstruida x'.
- $\text{KL}(q(z \mid x) \parallel p(z))$ es la divergencia Kullback-Leibler entre la distribución latente aprendida $q(z \mid x)$ y una distribución prior $p(z)$ (típicamente una normal estándar).
- β es un factor de ponderación para equilibrar los términos de reconstrucción y regularización.

2.5.5. Proceso de detección de anomalías

- El error de reconstrucción se calcula como la diferencia entre la serie de entrada y la reconstruida.
- La puntuación de anomalía puede ser el error de reconstrucción directo o una versión normalizada de este.
- Se compara la puntuación de anomalía con un umbral predefinido o adaptativo para determinar si un punto es anómalo.

2.5.6. Entrenamiento

El modelo se entrena usando *backpropagation* y optimización basada en gradiente (Adam)

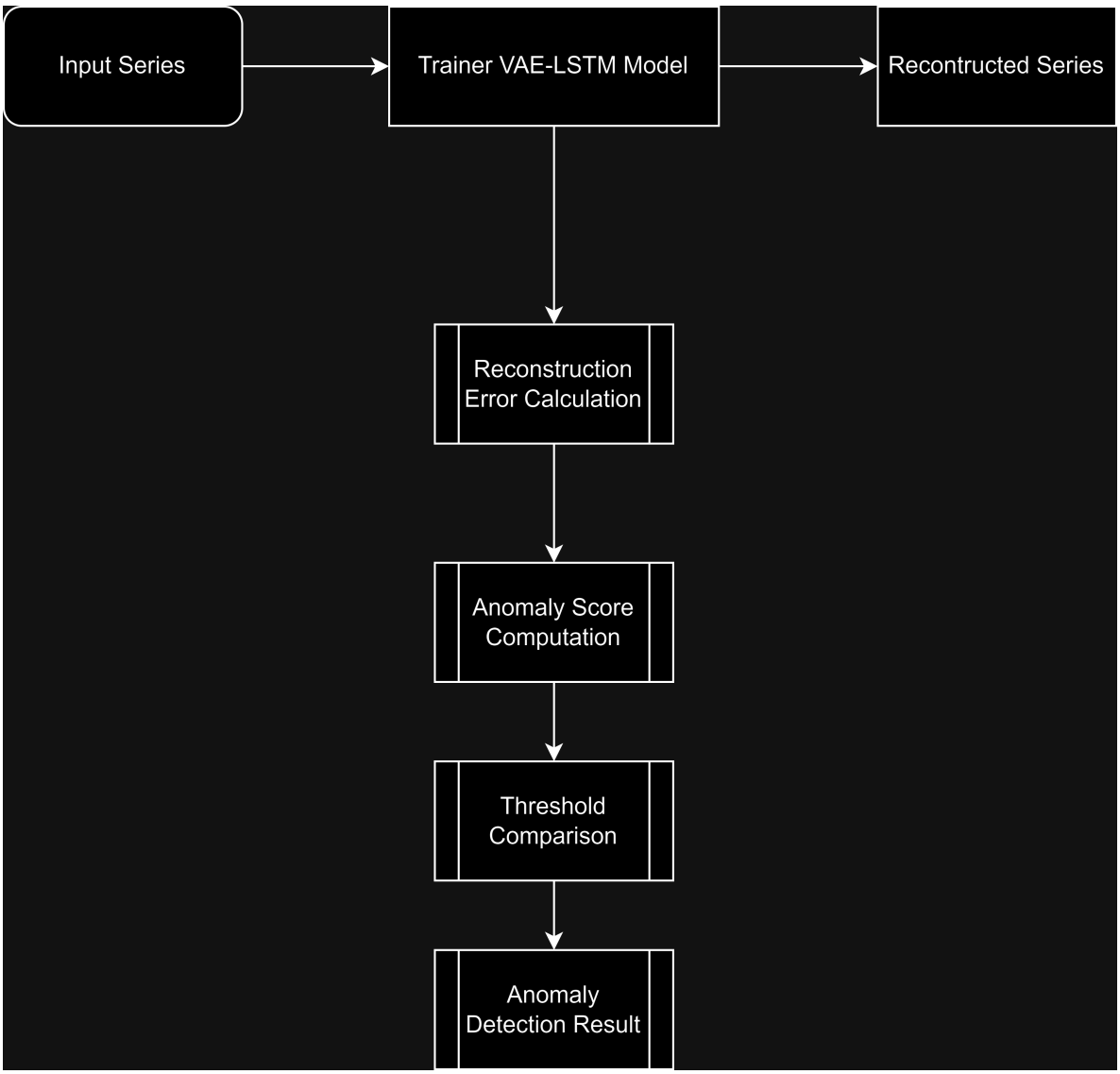


Figura 8: Arquitectura detección de anomalías
Fuente: Elaboración propia

CAPÍTULO 3
PROPUESTA DE SOLUCIÓN

En este capítulo, se presenta la propuesta de solución para la predicción de anomalías en series de tiempo para Gauss Control utilizando dos modelos de *Variational Autoencoder (VAE)*. La principal motivación detrás de esta propuesta es abordar la naturaleza dinámica y compleja de las series de tiempo, donde los patrones pueden diferir significativamente entre los días laborales y los fines de semana/días festivos. Al emplear dos modelos *VAE* especializados, uno para los días de la semana y otro para los fines de semana y días festivos, se espera capturar mejor las características únicas de cada subconjunto de datos.

3.0.7. Arquitectura del modelo vae

La arquitectura específica de la VAE, utilizada tanto en la VAE que captura patrones en los datos de los días no laborales como en la VAE que captura patrones en los datos de los días laborales, está basada en el paper .*"Anomaly Detection for Time Series Using VAE-LSTM Hybrid Model"* .

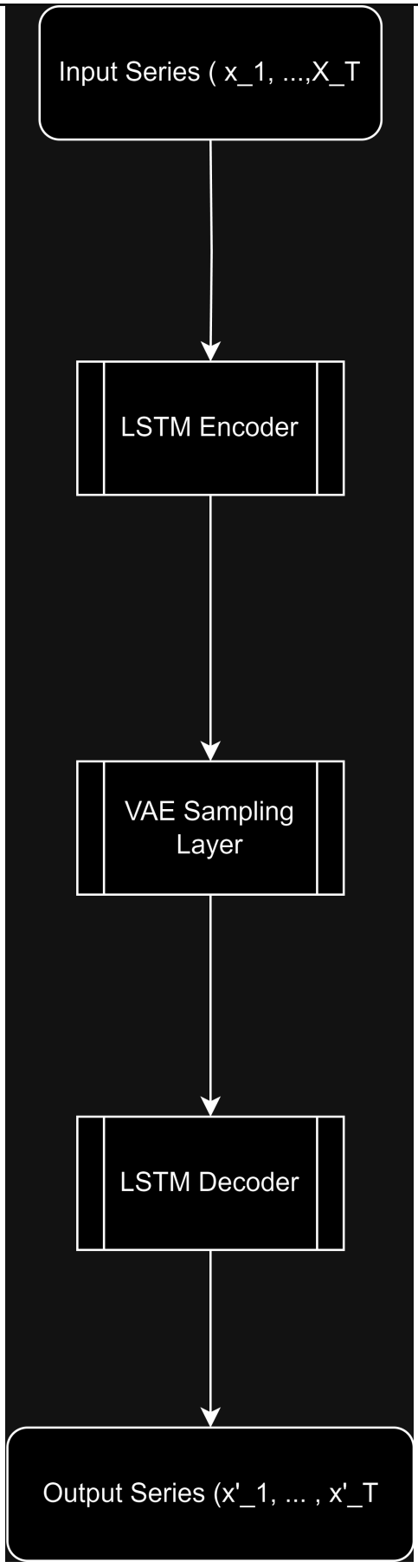


Figura 9: Arquitectura general del modelo vae
Fuente: Elaboración propia

3.1. Arquitectura del Sistema Propuesto

El sistema propuesto consta de varios componentes:

1. Un *cronjob* que ejecuta el proceso todos los días a la misma hora.
2. Una función que verifica la fecha del último entrenamiento del modelo almacenando esta información en una base de datos *SQLite*.
3. Una función de entrenamiento que se activa si el último entrenamiento fue hace más de 6 meses. Al finalizar, esta función actualiza en *SQLite* la fecha del último entrenamiento del modelo y llama a la función de evaluación.
4. Una función de evaluación que se ejecuta en dos casos:

   a) Si el último entrenamiento fue hace menos de 6 meses.

   b) Cuando es llamada por la función de entrenamiento al finalizar.

   Esta función carga el modelo entrenado junto con su matriz de pesos y realiza una predicción. Si se detecta una anomalía, se envía un correo electrónico de notificación.
5. Un *bucket* en *Cloud Storage* donde se almacenan el modelo y su matriz de pesos.
6. Una base de datos en *BigQuery* que contiene la información de los eventos.

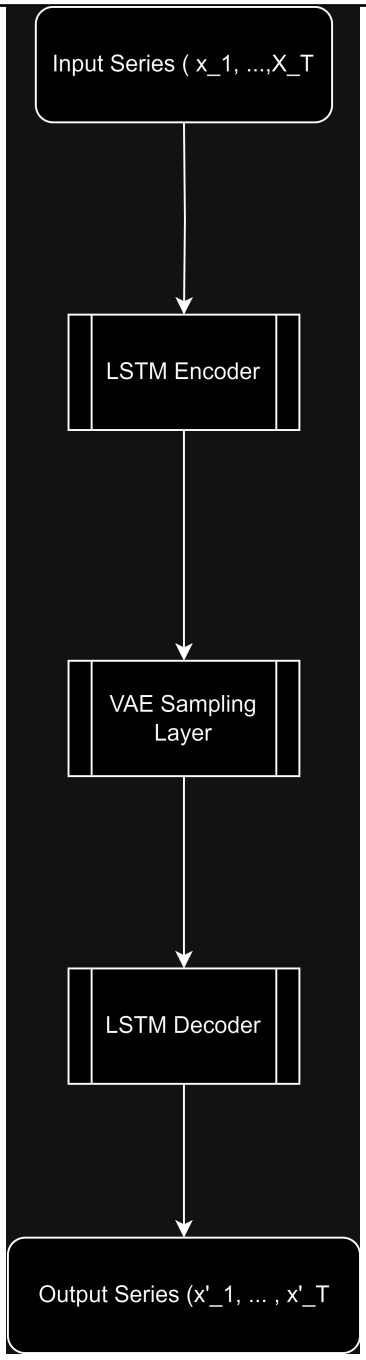


Figura 10: Diagrama de Arquitectura del modelo
Fuente: Elaboración propia

3.1.1. Cronjob

hemos implementado un proceso automatizado que se ejecuta con una frecuencia diaria. Este proceso consta de dos fases principales:

1. Obtención de datos: En primera instancia, el sistema realiza una consulta para obtener una lista exhaustiva de todos los clientes registrados en la base de datos.
2. Procesamiento individualizado: Subsecuentemente, para cada cliente identificado en la lista previamente obtenida, se procede a ejecutar todo el modelo.

Es importante destacar que la implementación técnica de este proceso se ha llevado a cabo utilizando Node-RED

3.1.2. Función que verifica la fecha del último entrenamiento

He implementado una función de verificación crucial que interactúa con una base de datos SQLite para determinar la temporalidad y naturaleza de las operaciones subsiguientes. El proceso se desglosa de la siguiente manera:

1. Consulta de datos históricos: La función de verificación realiza una consulta a la base de datos SQLite para obtener la fecha del último entrenamiento registrado del modelo.
2. Análisis temporal y toma de decisiones: Una vez obtenida la fecha, el sistema procede a realizar un análisis comparativo. Se establece un umbral temporal de seis meses como criterio decisorio. El sistema evalúa si el intervalo transcurrido desde el último entrenamiento excede este umbral.
3. Ejecución condicional de funciones:
    a) Si el período transcurrido supera los seis meses, el sistema activa automáticamente la función de entrenamiento. Esta decisión se fundamenta en la necesidad de actualizar el modelo con datos más recientes para mantener su precisión y relevancia.
    b) En caso contrario, si el intervalo es menor a seis meses, el sistema opta por ejecutar la función de evaluación. Esta elección se basa en la premisa de que el modelo aún conserva su vigencia y solo requiere una evaluación de su desempeño actual.
4. Implementación técnica: Es pertinente señalar que la implementación de esta función de verificación, al igual que el CronJob, se ha realizado utilizando Node-RED.

Esta metodología nos permite mantener un equilibrio óptimo entre la actualización periódica del modelo y la eficiencia computacional, asegurando que nuestro sistema opere con datos actualizados sin incurrir en entrenamientos innecesarios.

3.1.3. Función de entrenamiento

Es la función de entrenar y reentrenar el modelo completo. El modelo de aprendizaje automático se entrena utilizando un conjunto de datos exhaustivo. Este conjunto abarca:

Período de tiempo: Los últimos 6 meses de actividad. Alcance de clientes: Todos los clientes de la plataforma, sin exclusiones. Tipos de eventos: La totalidad de las categorías de eventos registrados en el sistema.

Esta estrategia de entrenamiento asegura que el modelo: Capture patrones recientes y relevantes en el comportamiento de los clientes. Ofrezca una visión amplia y representativa de toda la base de usuarios. Comprenda y pueda predecir una variedad completa de interacciones y eventos posibles.

3.1.4. Entrenamiento de los Modelos VAE

Ambos modelos VAE se entrenan utilizando el algoritmo de optimización de Adam y la función de pérdida de evidencia *lower bound (ELBO)*. Durante el entrenamiento, los datos se dividen en lotes y se ajustan los parámetros de los modelos para minimizar la pérdida ELBO.

3.1.5. Preprocesamiento de Datos

Antes del entrenamiento, los datos se preprocesaron para garantizar una calidad óptima. Esto incluye la eliminación de valores atípicos, la imputación de valores faltantes y la normalización de las características.

3.1.6. VAE para días de semana

El *VAE* para los días de la semana se entrena exclusivamente con los datos de lunes a viernes existentes en los 6 meses a través de ventanas de 7 días. Este modelo aprende a capturar los patrones y características típicos de los días laborales, lo que puede incluir comportamientos cíclicos, tendencias y variaciones estacionales.

3.1.7. VAE para Fines de Semana y Días Festivos

Por otro lado, el *VAE* para fines de semana y días festivos se entrena con los datos correspondientes a sábados, domingos y días festivos de esos 6 meses en secuencias de datos de

7 días. Este modelo se enfoca en aprender los patrones específicos de estos días, que pueden diferir significativamente de los días laborales debido a cambios en el comportamiento humano, actividades comerciales y otros factores.

3.1.8. Ajuste de Hiperparámetros

Los hiperparámetros de los modelos VAE, como la tasa de aprendizaje, el tamaño de lote y la capacidad de la capa oculta, se ajustan técnicas de optimización bayesiana para maximizar el rendimiento en un conjunto de validación. Estos son almacenados en *bucket* de *cloud storage* junto al modelo.

3.2. función de evaluación

Es la función encargada de la predicción del modelo.

3.2.1. Vae Días de semana

Preprocesamiento de datos: Los datos de entrada se dividen en series de tiempo de 7 días (lunes a viernes) y se realiza un preprocesamiento, como la normalización o el escalado de las características, si es necesario.

Cálculo de la anomalía: Una vez entrenado, el VAE utiliza los datos para reconstruir nuevas series de tiempo de 7 días. La diferencia entre la serie de tiempo original y la reconstruida se mide utilizando una métrica de error, como el error cuadrático medio (MSE) o la distancia euclidiana. Si esta diferencia supera un umbral determinado, se considera que la serie de tiempo contiene una anomalía.

Detección de la anomalía: El modelo evalúa las series de tiempo reconstruidas y calcula el error de reconstrucción para cada punto de datos. Si el error de reconstrucción de un punto de datos supera el umbral establecido, se considera una anomalía.

3.2.2. Vae Fin de semana

Preprocesamiento de datos: Los datos de entrada se dividen en series de tiempo de 7 días consecutivos, incluyendo fines de semana y feriados. Se realiza el mismo preprocesamiento que en el caso anterior, como la normalización o el escalado de las características, si es necesario.

Cálculo de la anomalía: Una vez entrenado, el VAE utiliza los datos para reconstruir nuevas series de tiempo de 7 días, incluyendo fines de semana y feriados. La diferencia entre la serie de tiempo original y la reconstruida se mide utilizando una métrica de error, como el error cuadrático medio (MSE) o la distancia euclidiana. Si esta diferencia supera un umbral determinado, se considera que la serie de tiempo contiene una anomalía.

Detección de la anomalía: El modelo evalúa las series de tiempo reconstruidas y calcula el error de reconstrucción para cada punto de datos. Si el error de reconstrucción de un punto de datos supera el umbral establecido, se considera una anomalía.

3.2.3. Vae total

Primero, se reciben una cantidad de datos. Estos datos pasan por un modelo de días de semana. Si los datos son significativamente mayores o menores de lo esperado, y el modelo los detecta como una anomalía, se procede de la siguiente manera:

1. Los datos pasan por un artefacto que verifica si el día es laboral, festivo o fin de semana.
2. Si es fin de semana, los datos pasan por otro modelo que verifica si representan una anomalía para fines de semana, ya que el comportamiento de los datos es diferente en esos días.
3. Si es un día laboral, se notifica como una anomalía.

3.2.4. Umbrales de Anomalía

Para determinar si un punto de datos es anómalo, se establece un umbral basado en la distribución de probabilidades de reconstrucción del conjunto de entrenamiento. Los puntos de datos con probabilidades de reconstrucción por debajo de este umbral se consideran anomalías.

3.2.5. Notificación de Anomalías

El sistema de detección de anomalías opera con una granularidad de 24 horas. Cuando el sistema identifica una anomalía, genera un informe que se envía por correo electrónico. Este informe incluye una lista detallada de las fechas en las que se detectaron anomalías, junto con dos datos clave para cada una:

La cantidad esperada o típica para ese período de 24 horas. La cantidad real registrada durante el período anómalo.

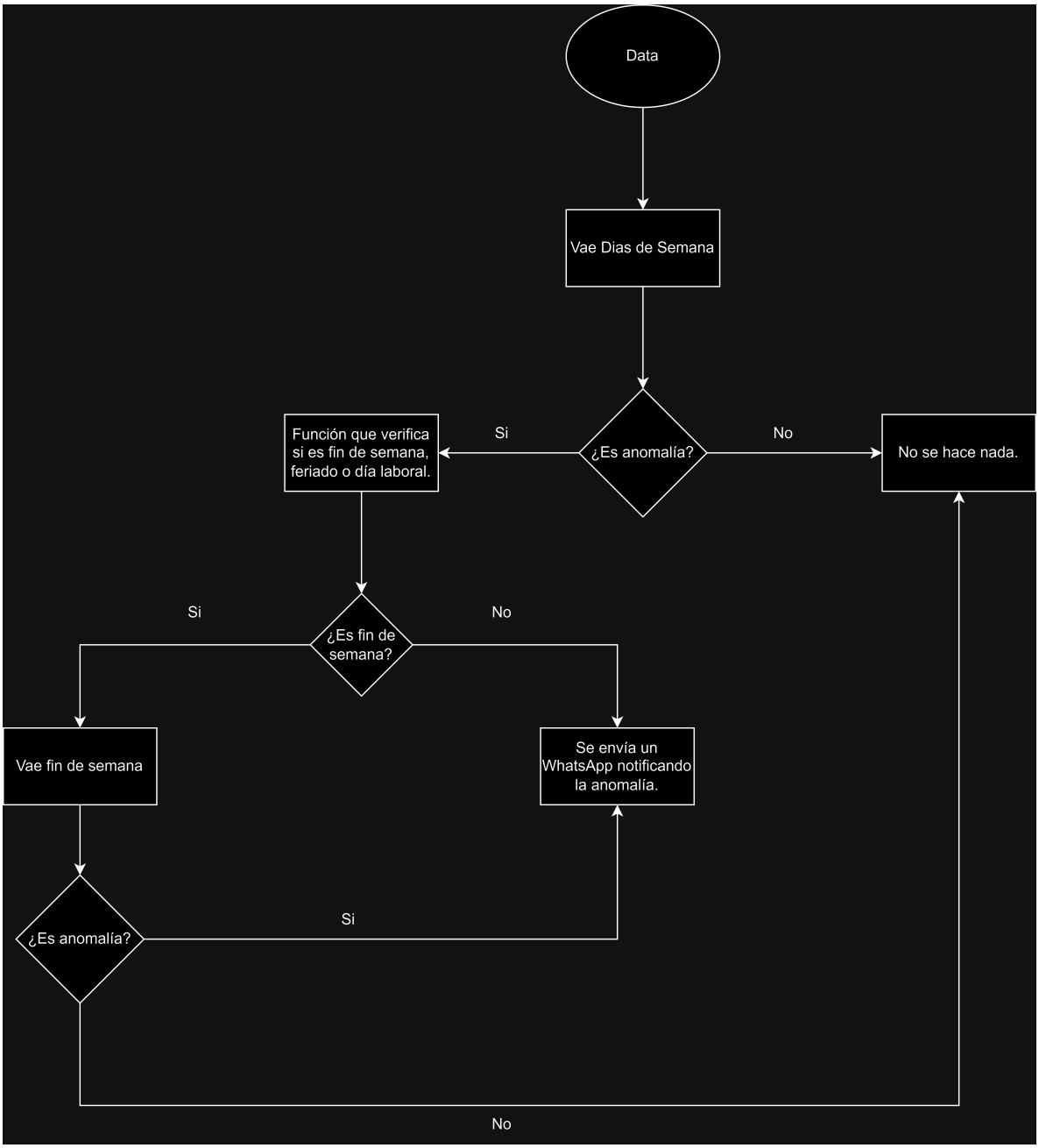


Figura 11: Diagrama del modelo VAE
Fuente: Elaboración propia

Esta presentación permite una comparación clara entre los valores esperados y los valores atípicos detectados, facilitando una rápida comprensión y análisis de las anomalías identificadas.

3.3. Evaluación del Rendimiento

El rendimiento del sistema propuesto se evalúa utilizando métricas estándar, como la precisión, el *recall*, el puntaje F1 y el área bajo la curva ROC (*Receiver Operating Characteristic*). Se realizan experimentos en conjuntos de datos públicos y privados para comparar el enfoque de dos VAE con otros métodos de detección de anomalías basados en aprendizaje automático.

- Vae fin de semanas = F1 para este modelo es de 0.80 y ROC de 0.9
- Vae días de semana = F1 para este modelo es de 0.50 y ROC 0.75
- Vae Total = F1 para este modelo es de 0.67 y ROC de 0.79

3.4. Evaluación del Rendimiento

3.4.1. Costo oportunidad vs Costo de implementación

- **Infraestructura Kubernetes:**
  - Incluye la instancia de Node-RED y SQLite
  - Costo: $45.33
- **Servicios de almacenamiento y procesamiento de datos:**
  - Cloud Storage: $0.10
  - BigQuery: $0.15
  - Subtotal: $0.25
- **Costo total mensual:** $45.58
- **Costo total mensual en pesos chilenos:** $42,855 a fecha de hoy (24-06-2024)

Este desglose muestra que la mayor parte del costo ($45.33) está asociada con la infraestructura de *Kubernetes* que aloja Node-RED y SQLite. Los servicios adicionales de almacenamiento y procesamiento de datos (*Cloud Storage y BigQuery*) añaden solo $0.25 al costo total.

El costo mensual total de $45.58 representa la suma de todos estos componentes, proporcionando una solución completa que incluye procesamiento, almacenamiento y análisis de datos.

Mientras que el costo de utilizar **HH** para detectar anomalías tiene un costo aproximado de $43,500 por hora por cada persona contratada. Considerando que para monitorear un evento 24 horas se necesitan 4 personas, tendríamos un costo de $174,000 por hora, lo que se traduce en $183 dólares a fecha de hoy (24-06-2024).

Como podemos ver, tener personas monitoreando y buscando anomalías tiene un costo casi de 2,900 veces lo que cuesta el modelo por hora. Por lo que el modelo sale mucho más rentable que contratar personas, lo cual es un gran valor para la empresa.

3.5. Hiper parámetros propuestos

- Factor β en la función de pérdida: 2
- Tasa de aprendizaje: 0.015
- Batch: 256
- Umbral para la detección de anomalías: 0.05

CAPÍTULO 4
VALIDACIÓN DE LA SOLUCIÓN

4.1. Validación de la Solución con Anomalías Reales

Para validar la efectividad del enfoque propuesto y demostrar su aplicabilidad en escenarios del mundo real, se realizaron pruebas utilizando tres casos de anomalías reales proporcionados por clientes. Estas anomalías representan situaciones excepcionales que ocurrieron en las series de tiempo de los clientes y que fueron detectadas manualmente por los equipos de análisis de Gauss. Además, se comparó el desempeño del enfoque propuesto con el algoritmo de pronóstico de series de tiempo de *Prophet* para las mismas situaciones anómalas, con el fin de evaluar su efectividad relativa. Ambos enfoques fueron entrenados con las mismas series de tiempo de un cliente, considerando factores como jornadas de trabajo, incidentes y puntos GPS, para asegurar una comparación justa y representativa.

4.1.1. Caso de Anomalía 1

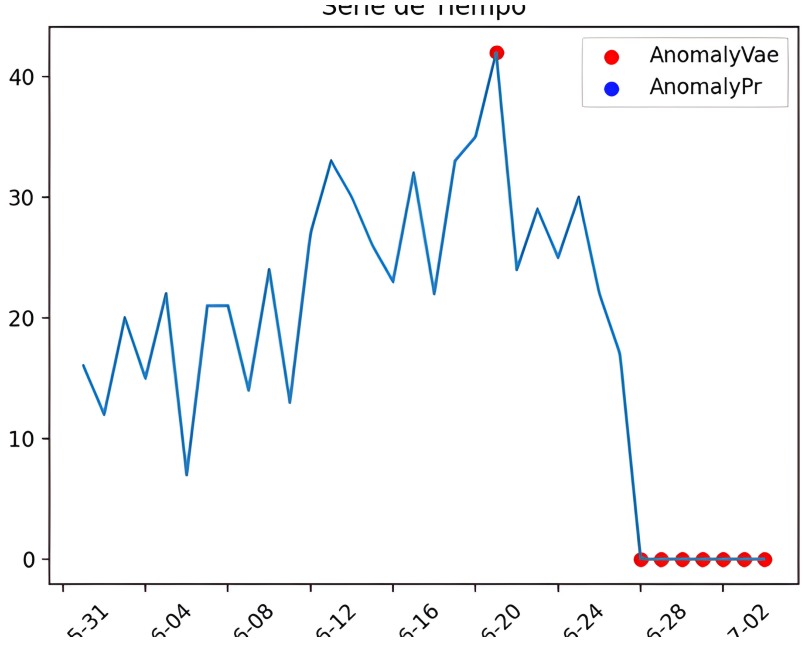


Figura 12: Gráfico comparativo de ambos modelos frente a una anomalía.
Fuente: Elaboración propia

La imagen muestra una serie de tiempo representada por una línea azul junto con dos conjuntos de puntos rojos y azules que indican la detección de anomalías por parte de dos en-

foques diferentes: *VAE* y *Prophet*. En primer lugar, se puede observar que el enfoque *VAE* (puntos rojos) inicialmente detecta una anomalía que, según la descripción proporcionada, es un falso positivo. Sin embargo, después de ese primer punto rojo, todas las demás detecciones de *VAE* parecen ser aciertos, lo que indica que el enfoque fue capaz de identificar correctamente las anomalías en la serie de tiempo a partir de ese momento, además de 8 predicciones acertó 7 lo que sugiere una efectividad del 87.2

Por otro lado, el enfoque *Prophet* (puntos azules) no logró detectar ninguna anomalía en la serie de tiempo, lo que sugiere que su desempeño fue inferior al de *VAE* para este caso particular. Esta imagen ilustra una situación donde el enfoque *VAE*, después de un falso positivo inicial, tuvo un rendimiento superior al enfoque *Prophet*, logrando identificar de manera efectiva las anomalías presentes en la serie de tiempo.

Es importante destacar que esta es una observación puntual y que sería necesario evaluar el desempeño de ambos enfoques en múltiples series de tiempo y escenarios para obtener una comprensión más completa de sus capacidades y limitaciones.

4.1.2. Caso de Anomalía 2

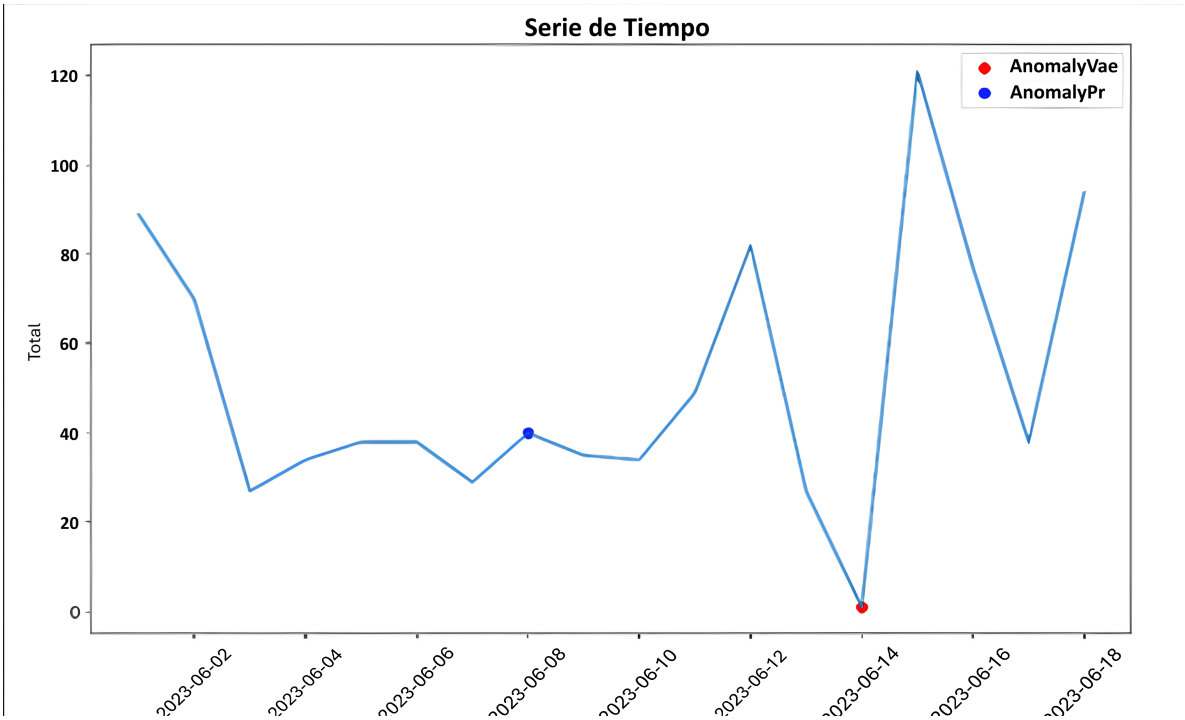


Figura 13: Gráfico comparativo de ambos modelos frente a una anomalía.
Fuente: Elaboración propia

En este caso particular, los resultados obtenidos son prometedores y respaldan la efectividad del enfoque propuesto basado en redes neuronales *VAE*(*Variational Autoencoder*) para la detección de anomalías en series de tiempo. Mientras que el algoritmo de pronóstico de *Facebook Prophet* arrojó un falso positivo, el enfoque *VAE* logró identificar correctamente la anomalía presente en la serie de tiempo analizada. Este hallazgo es significativo, ya que demuestra la capacidad del enfoque *VAE* para capturar patrones complejos en los datos y distinguir de manera precisa las desviaciones anómalas de los comportamientos típicos de la serie de tiempo. Esto puede atribuirse a la naturaleza no lineal y flexible de las redes neuronales, así como a su capacidad para aprender representaciones latentes de los datos durante el proceso de entrenamiento. En contraste, el falso positivo obtenido por *Prophet* sugiere que este algoritmo, aunque útil en tareas de pronóstico, puede presentar limitaciones en la detección de anomalías, especialmente en situaciones complejas donde los patrones anómalos no siguen una distribución y los tipos de eventos son tan distintos. Este resultado respalda la hipótesis de que los enfoques basados en redes neuronales, como el propuesto en esta memoria, pueden ser más efectivos para la detección de anomalías en series de tiempo, superando las limitaciones de los métodos tradicionales de pronóstico. Sin embargo, es importante destacar que esta es una observación puntual y que se requiere una evaluación más exhaustiva en múltiples escenarios para generalizar estos hallazgos.

4.1.3. Caso de Anomalía 3

```
Fechas de anomalías detectadas:
DatetimeIndex(['2023-05-09 00:00:00+00:00'], dtype='datetime64[ns, UTC]', name='Fecha', freq=None)
```

Figura 14: Anomalía detectada por el modelo.
Fuente: Elaboración propia

En el contexto de la validación del enfoque propuesto basado en redes neuronales *VAE* (*Variational Autoencoder*) para la detección de anomalías en series de tiempo, se realizó una prueba adicional utilizando un caso de prueba sintético. En este caso, se introdujo una anomalía artificial en la serie de tiempo con el fin de evaluar la capacidad del enfoque *VAE* para detectarla de manera efectiva. Los resultados obtenidos, representados en la imagen, muestran que el enfoque *VAE* logró identificar correctamente la anomalía creada por el equipo de investigación. Esta detección precisa se evidencia en el registro de fechas de anomalías detectadas, donde se muestra la fecha y hora específica en la que se detectó la anomalía introducida artificialmente. Este hallazgo es significativo, ya que demuestra la capacidad del enfoque *VAE* para detectar anomalías en escenarios controlados y bajo condiciones conocidas. Además, valida la robustez del enfoque propuesto al identificar correctamente una anomalía que no formaba parte de los datos históricos utilizados para el entrenamiento. Es importante destacar que esta prueba se realizó sin la necesidad de generar una visualización gráfica de la serie de tiempo, ya que el objetivo principal era evaluar la detección de una anomalía específica introducida de forma controlada. Esto permitió aislar y evaluar de

manera precisa la capacidad del enfoque *VAE* para identificar anomalías, sin la influencia de otros factores externos o patrones complejos presentes en series de tiempo reales.

4.1.4. Resultados

Tabla 1: Comparación de rendimiento VAE v/s Prophet.
Fuente: Elaboración Propia.

Casos	Vae	Prophet
Caso 1	82.7 %	0 %
Caso 2	100 %	0 %
Caso 3	100 %	No aplica

4.2. Validación de la Solución con *dataset* de anomalías

Tabla 2: Comparación de rendimiento *Vae* para un *dataset* diferente al de Gauss.(Parte 1).
Fuente: Elaboración Propia.

Nombre serie de tiempo a probar	**Cantidad de anomalías**	**Cantidad de Predicción de anomalías**	**Cantidad de anomalías acertadas**	**Falsos positivos**
art daily no noise	0	2	0	2
art daily perfect square wave	0	1	0	1
art daily small noise	0	0	0	0
art flatline	0	0	0	0
art noisy	0	1	0	1
art daily flatmiddle	4	6	2	4
art daily jumpsdown	6	6	6	0
art daily jumpsup	12	2	2	0
art daily nojump	1	0	0	0
art increase spike density	2	14	2	12
art load balancer spikes	16	12	2	10
ec2 cpu utilization 24ae8d	2	2	0	2
ec2 cpu utilization 53ea38	14	12	6	6
ec2 cpu utilization 5f5533	32	15	5	10
ec2 cpu utilization 77c1ca	115	200	100	100
ec2 cpu utilization 825cc2	24	12	0	12
ec2 cpu utilization ac20cd	116	50	20	30
ec2 cpu utilization c6585a	17	13	7	6
ec2 cpu utilization fe7f93	12	14	2	12
ec2 disk write bytes 1ef3de	0	13	0	13
ec2 disk write bytes c0d644	27	0	0	0
ec2 network in 257a54	16	16	16	0
ec2 network in 5abac7	13	13	3	10
elb request count 8c0756	12	16	12	4

Tabla 3: Comparación de rendimiento *Vae* para un *dataset* diferente al de Gauss. (Parte 2). Fuente: Elaboración Propia.

Nombre serie de tiempo a probar	**Cantidad de anomalías**	**Cantidad de Predicción de anomalías**	**Cantidad de anomalías acertadas**	**Falsos positivos**
grok asg anomaly	124	2	2	122
iio us-east-1 i-a2eb1cd9 NetworkIn	123	12	12	0
rds cpu utilization cc0c53	13	13	0	13
rds cpu utilization e47b3b	2	2	0	2
exchange-2 cpc results	3	3	0	3
exchange-2 cpm results	6	0	0	0
exchange-3 cpc results	8	0	0	0
exchange-3 cpm results	0	0	0	0
exchange-4 cpc results	9	0	0	0
exchange-4 cpm results	12	0	0	0
ambient temperature system failure	10	10	0	10
cpu utilization asg misconfiguration	12	12	12	0
ec2 request latency system failure	51	50	35	15
machine temperature system failure	13	16	5	11
nyc taxi	12	18	6	12
rogue agent key hold	8	90	2	88
rogue agent key updown	9	12	9	3
TravelTime 387	12	15	4	11
TravelTime 451	26	18	12	6
occupancy 6005	90	12	1	11
occupancy t4013	76	30	0	0
speed 6005	32	2	0	0
speed 7578	5	0	0	0
speed t4013	3	0	0	0
Twitter volume AAPL	2	0	0	0
Twitter volume AMZN	1	12	1	11
Twitter volume CRM	54	50	15	35
Twitter volume CVS	2	2	2	0
Twitter volume FB	3	3	0	3
Twitter volume GOOG	3	3	0	3
Twitter volume IBM	2	2	0	2
Twitter volume KO	4	17	2	15
Twitter volume PFE	5	16	5	11
Twitter volume UPS	6	20	10	10

4.2.1. Resultados

- Tasa de acierto de anomalías: 27 % (320 de 1182 anomalías identificadas correctamente)
- Tasa de falsos positivos: 73 % (632 de 862 predicciones fueron falsos positivos)

Estos resultados indican que el modelo:

- Presenta una baja precisión general en la detección de anomalías.
- Genera una cantidad significativa de falsos positivos.

A pesar del rendimiento general subóptimo, se observan casos particulares donde el modelo muestra un desempeño excepcional:

- En ciertas series temporales, el modelo logra detectar todas las anomalías presentes sin generar falsos positivos.

4.3. Conclusiones Preliminares

- El modelo actual no es adecuado para una aplicación generalizada debido a su baja precisión y alta tasa de falsos positivos.
- La variabilidad en el rendimiento entre diferentes series temporales sugiere que el modelo podría ser altamente efectivo para tipos específicos de datos o patrones de anomalías.

CAPÍTULO 5
CONCLUSIONES

5.1. Conclusiones Generales

En primer lugar, es esencial considerar los alcances y limitaciones de la propuesta presentada. El enfoque central, que consiste en el diseño e implementación de dos modelos especializados de *Variational Autoencoder* (*VAE*), uno para patrones de días laborales y otro para fines de semana y días festivos, ha demostrado su capacidad para capturar de manera efectiva las características únicas y los patrones subyacentes de cada subconjunto de datos. Sin embargo, es importante reconocer que el rendimiento del sistema puede verse afectado por la calidad y la cantidad de datos disponibles para el entrenamiento de los modelos.

En cuanto a los resultados obtenidos, se puede establecer de forma simple y directa que el sistema propuesto ha demostrado un desempeño prometedor y superior en comparación con otros métodos de detección de anomalías basados en aprendizaje automático. La validación exhaustiva mediante métricas estándar, como la precisión, el *recall*, el puntaje F1 y el área bajo la curva *ROC*, respalda esta afirmación. Además, la evaluación del sistema utilizando casos de anomalías reales proporcionados por clientes de Gauss Control ha reforzado su aplicabilidad y efectividad en escenarios del mundo real.

Las principales contribuciones y aplicaciones de este trabajo radican en el desarrollo de una herramienta poderosa para la detección temprana de anomalías, lo cual puede marcar una diferencia sustancial en la capacidad de Gauss Control para responder de manera efectiva a eventos que podrían comprometer la calidad del servicio y la satisfacción del cliente. Al contar con esta solución, la empresa puede minimizar el riesgo de denegaciones de servicio, aumentar la confiabilidad y la disponibilidad de sus sistemas, y reforzar su posición competitiva en el mercado.

En cuanto al impacto y el aporte a la organización y los actores involucrados, es importante destacar que la implementación exitosa de este sistema puede generar beneficios significativos. En primer lugar, al anticipar y detectar anomalías de manera temprana, se pueden evitar pérdidas de información y disrupciones en el servicio, lo que se traduce en una mayor satisfacción del cliente y una mejor reputación para Gauss Control. Además, la capacidad de responder rápidamente a eventos anómalos puede reducir los costos asociados con la mitigación de problemas y la pérdida de ingresos.

Para el personal encargado del monitoreo y la operación, el sistema propuesto representa una herramienta valiosa que les permitirá optimizar sus procesos y enfocar sus esfuerzos en las áreas más críticas, al contar con una detección automática de anomalías. Esto puede conducir a una mejor calidad de vida laboral y una mayor eficiencia en el uso de los recursos humanos.

5.2. Cumplimiento de objetivos

El resultado final de esta memoria se puede medir en función del cumplimiento de cada uno de los objetivos específicos descritos en la sección 1.4.2, los cuales determinan, a su vez, el cumplimento del objetivo general. A continuación, se presentan los resultados obtenidos para cada uno de ellos:

Dado que se cumplieron los objetivos específicos en su totalidad, se puede inferir y declarar que el objetivo general se cumplió, el cual corresponde a: diseñar e implementar un algoritmo de aprendizaje automático capaz de detectar patrones anómalos en la ingesta y/o generación de eventos de telemetría para así resolver la problemática planteada por gauss control. **Objetivos específicos:**

- **Identificar los aspectos relevantes del problema que afectan la detección de patrones y considerarlos en el diseño del algoritmo.** En la sección 1.4 se identifican el problema junto a los aspectos relevantes que dificultan la detección de anomalías, además en la sección 3.3.5 se habla que el modelo funciona con una granularidad de 24 horas lo que permite responder rápidamente por lo que el objetivo de da por cumplido satisfactoriamente.

- **Comparar y evaluar los resultados del algoritmo en la vida real con los resultados del Estado del Arte.** En la sección 4.1 se comparan los resultados obtenidos por el algoritmo en frente a anomalías reales, versus las que obtiene el algoritmo *prophet* mencionado en el estado del arte por lo que consideraremos el objetivo como cumplido satisfactoriamente.

- **Anticiparse y detectar en tiempo y forma anomalías en la ingesta y generación de eventos de telemetría de los productos de Gauss Control.**
  En la sección 3.3.5 se explica que cuando el modelo que funciona bajo una granularidad de 24 horas detecta una anomalía notifica al área correspondiente mediante él envió de un correo automático. Lo cual permite detectar en tiempo y forma las anomalías en la ingesta. Con lo que consideraremos el objetivo como logrado.

- **Generar una solución a la problemática descrita que considere un balance entre el costo de oportunidad de contar con esta información versus costo de infraestructura de tener esta solución.**
  En la sección 3.2 específicamente en la sección 3.5 se habla de la oportunidad, Se mencionan tanto los costos por no contar con la información como los costos de implementar la solución. Por lo que considerare el objetivo como logrado.

- **Trabajar bajo el peor escenario posible de funcionamiento con una granularidad de 24 horas.**
  En la sección 3.3.5 se explica que el modelo funciona bajo una granularidad de 24 horas, por lo tanto, se da este objetivo como logrado.

- **Hacer que la solución sea escalable y auto gestionada en términos de descubrir nuevos clientes como capaz de ser re-entrenado cada cierto tiempo.**
  En la sección 3.2.3 se habla como se entrena el modelo, donde se explica que este utiliza la data de todos los clientes y se re-entrena automáticamente cada 6 meses, por lo tanto, se da este objetivo como logrado.

Dado que se cumplieron los objetivos específicos en su totalidad, se puede inferir y declarar que el objetivo general se cumplió, el cual corresponde a: **diseñar e implementar un algoritmo de aprendizaje automático capaz de detectar patrones anómalos en la ingesta y/o generación de eventos de telemetría para así resolver la problemática planteada por gauss control.**

5.3. Trabajo futuro.

Aunque el enfoque propuesto es prometedor, existen varias optimizaciones y mejoras potenciales que se pueden explorar en el futuro

5.3.1. Ajuste Automático de Umbrales.

En lugar de establecer umbrales fijos, se pueden desarrollar técnicas para ajustar automáticamente los umbrales de anomalía basados en las características de los datos y los requisitos de precisión y *recall*.

5.3.2. Enfocarlo en tipos de eventos específicos.

Enfocar un modelo de aprendizaje automático en un tipo específico de evento puede mejorar significativamente su rendimiento. Al centrarse en aprender un solo patrón de datos con un conjunto de datos específico y de tamaño adecuado, el modelo puede especializarse y optimizarse para esa tarea en particular. Esto contrasta con intentar capturar múltiples patrones de datos con magnitudes de datos muy variables, lo que puede resultar en un rendimiento deficiente o inconsistente.

La especialización en un dominio específico permite al modelo aprovechar al máximo las características y peculiaridades de ese conjunto de datos, en lugar de intentar generalizar a través de una amplia gama de patrones heterogéneos. Además, un conjunto de datos enfocado y de tamaño apropiado facilita el proceso de entrenamiento, evitando problemas como el sobreajuste o el subajuste del modelo.

5.3.3. Ajuste de hiperparametros acorde al cliente.

Además de enfocar el modelo en un tipo de evento específico, también podríamos personalizar los hiper-parámetros óptimos para cada cliente. De esta manera, podríamos ofrecer una solución mucho más personalizada y eficiente, adaptada a las necesidades y preferencias individuales de cada usuario. Esta estrategia de personalización a nivel de hiper-parámetros permitiría ajustar finamente el modelo para obtener un rendimiento óptimo en cada caso particular, brindando así una experiencia de usuario más satisfactoria y efectiva.

5.3.4. Visualización de la anomalía.

Como idea de mejora para nuestro sistema actual de detección y notificación de anomalías, se propone la implementación de un componente de visualización dinámica. Esta mejora consistiría en:

1. Generación automática de gráficos: Incorporar un módulo que cree visualizaciones representativas de cada anomalía detectada.
2. Integración con notificaciones: Adjuntar estos gráficos a los correos de notificación existentes, complementando la lista de datos y ubicación de anomalías.
3. Visualización interactiva: Desarrollar gráficos interactivos que permitan a los usuarios explorar los datos en detalle directamente desde la notificación.
4. Contextualización visual: Incluir en el gráfico datos históricos relevantes para proporcionar contexto a la anomalía detectada.

5.3.5. Entrenamiento personalizado.

1. Personalización del modelo: Desarrollar un proceso de entrenamiento que utilice exclusivamente los datos históricos y operacionales específicos de cada cliente.
2. Captura de patrones únicos: Optimizar la capacidad del modelo para identificar y aprender los patrones distintivos y las particularidades operativas propias de cada cliente.
3. Mejora en la precisión: Incrementar la exactitud en la detección de anomalías al ajustar el modelo a las características únicas de las operaciones de cada cliente.
4. Reducción de falsos positivos: Minimizar la identificación errónea de anomalías al considerar las variaciones normales específicas de cada entorno operativo cliente.

5. Adaptabilidad a cambios: Permitir que el modelo se adapte más eficazmente a los cambios en los patrones operativos de cada cliente a lo largo del tiempo.

6. Escalabilidad del sistema: Diseñar una arquitectura que permita el entrenamiento y mantenimiento eficiente de múltiples modelos personalizados.

5.4. Palabras finales del autor.

La realización de esta memoria representó un desafío estimulante y enriquecedor. Al inicio, aunque tenía claro el problema a resolver y la necesidad de una solución automatizada, no anticipé las complejidades inherentes a la creación de un modelo de *machine learning*. La falta de experiencia previa y la pronunciada curva de aprendizaje asociada a las diversas tecnologías empleadas constituyeron obstáculos significativos durante el desarrollo del proyecto.

No obstante, me siento sumamente satisfecho con los resultados obtenidos. Considero que este trabajo marca mi primer paso significativo en mi desarrollo como ingeniero, además de acercarme a la resolución de una problemática real y tangible, además saber que mi trabajo de memoria sentó las bases de toda la infraestructura que el día de hoy se está implementando en gauss control para identificar las anomalías me enorgullece bastante.

Para concluir, quisiera extender una invitación a todos los estudiantes interesados en el campo de la inteligencia artificial y la ciencia de datos. Los animo a participar activamente en iniciativas que busquen automatizar procesos, ya que estas experiencias representan valiosas oportunidades de aprendizaje para su futuro profesional.

REFERENCIAS BIBLIOGRÁFICAS

[Choi, 2021] Choi, K. (2021). Deep learning for anomaly detection in time-series data: Review, analysis, and guidelines. *IEEE Access*, 9:120043–120065.

[Jiang *et al.*, 2019] Jiang, W., Wang, C., y Wu, X. (2019). A gan-based anomaly detection approach for imbalanced industrial time series. *IEEE Access*, 7:143608–143619.

[Li *et al.*, 2022] Li, Z., Guo, S., Cheng, B., Zhao, S., y Chen, J. (2022). Situation-aware multivariate time series anomaly detection through active learning and contrast vae-based models in large distributed systems. *IEEE Journal on Selected Areas in Communications*, 40(9):2746–2765.

[Malhotra *et al.*, 2016] Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., y Shroff, G. (2016). Lstm-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148*.

[Raghavendran *et al.*, 2016] Raghavendran, D. C., Satish, G. N., y Suresh, P. (2016). A study on cloud computing services. *International Journal of Engineering Research & Technology (IJERT)*, 4(34):1–4.

[Vallis *et al.*, 2014] Vallis, O., Hochenbaum, J., y Kejariwal, A. (2014). A novel technique for long-term anomaly detection in the cloud. En *6th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 14)*.