Mejorar la calidad del dato de la empresa a través de modelos de Machine Learning para mejorar las predicciones a futuro en la empresa

Thomas Andrés Düring Adriazola
tduring@alumnos.uai.cl
20.287.307-3

Pasantía

como requisito parcial para optar al título de la carrera de

Ingeniería Civil Informática

Supervisor a cargo
Ricardo Seguel
Profesor Guía
Nicolás Cenzano

Diciembre 2023

1. Resumen Ejecutivo

We Techs es una Empresa B que se dedica a la gestión eficiente del agua y otros fluidos industriales, basando su trabajo en valores como la sostenibilidad, la transparencia, la eficiencia y la innovación.

La calidad de los datos se ha vuelto un activo crucial para todas las empresas, ya que la toma de decisiones basada en datos imprecisos, anómalos o incorrectos puede acarrear grandes consecuencias. Durante la pasantía, se trabajó en mejorar la calidad de los datos de la empresa, dado que We Techs carece de un método para detectar datos anómalos y, por ende, no tiene la opción de documentarlos para comprender futuras anomalías. Esto se debe a que al trabajar específicamente en el área de TI y analítica avanzada, el desconocimiento de datos anómalos dificulta la precisión de las predicciones, ya que estos pueden distorsionar el comportamiento de los modelos.

Por tanto, se aplicaron modelos de detección de anomalías como Isolation Forest, Local Outlier Forest, LSTM Autoencoder, entre otros, con el fin de identificar y anticiparse a datos anómalos. Esta acción busca mejorar la precisión de los modelos predictivos.

Además, se creó una alarma que envía alertas por correo electrónico para notificar sobre anomalías en pozos de agua, indicando la fecha en que ocurrieron.

2. Abstract

We Techs is a B Corporation dedicated to efficient management of water and other industrial fluids, operating based on values of sustainability, transparency, efficiency, and innovation.

Data quality has become a critical asset for all companies. Decision-making based on imprecise, anomalous, or incorrect data can lead to significant consequences within an organization.

Hence, during the internship, the focus was on enhancing the company's data quality. We Techs lacks a method to detect anomalous data, which hinders their ability to document such anomalies and understand their causes in the future. This becomes a challenge particularly in the field of IT, specifically in advanced analytics, where working with predictive models without knowledge of present anomalies complicates predictions due to the lack of clear data behavior.

Therefore, anomaly detection models like Isolation Forest, Local Outlier Forest, LSTM Autoencoder, among others, were applied. The aim is to identify anomalous data proactively, aiding in the creation of more precise predictive models.

Additionally, an alert system was developed, triggering an email notification whenever an anomaly occurs in a water well, providing details about the anomaly and the date it occurred.

Índice

3. Introducción

Hoy en día, en el mundo empresarial los datos son lo mas valioso y fundamental para poder tener éxito. La calidad de la información se convirtió en un activo critico para poder tomar decisiones de la manera mas eficaz posible.
Dentro de este contexto, We Techs es una empresa B, la cual se dedica a ser aliados estratégicos en la gestión eficiente del agua y otros fluidos industriales, como por ejemplo, el uso de agua en pozos mineros, en centros de energía, centros sanitarios y agua potable rural. Esto lo logran a través del monitoreo, control remoto y análisis de datos, impulsando una operación sostenible y productiva.
We Techs trabaja con los siguientes valores, los cuales tienen publicados en su página:

- **Sostenibilidad:** Proveen información integral de la operación a los clientes para que puedan tomar decisiones en orden, para generar un impacto positivo social y ambiental.
- **Transparencia:** La obtención de datos en tiempo real permite cumplir con la normativa de cara al ente regulador, la opinión publica y comunidades que están involucradas en cada uno de los proyectos construyendo así relaciones basadas en la confianza en el largo plazo.
- **Eficiencia:** La agilidad es fundamental para asegurar la operación de los clientes. El servicio permite detectar anomalías de manera temprana lo que permite fortalecer la estabilidad y continuidad de la operación (este punto es el que actualmente esta con incertidumbre y de eso tratara el proyecto).
- **Innovación:** La cultura de aprendizaje y mejora continua permite entregar a los clientes una propuesta de valor integral de calidad y con la tecnología de punta que posibilite la mejora en la operación.

We Techs reciba una gran cantidad de datos a analizar. Y la cantidad de datos recibidos cada vez sera mas grande, lo que genera una preocupación que es muy importante no solo para We Techs si no que para todas las empresas que manejen grandes cantidades de datos, esto es ***la calidad del dato***[1].

[1]La calidad del dato se refiere a que tan confiable, precisa y útil es la información que se utiliza que se utiliza para la toma de decisiones

3.1. Contexto del problema y oportunidad

La toma de decisiones basada en datos anómalos, imprecisos o incorrectos puede tener grandes consecuencias, que van desde ineficiencias en la operación hasta posibles perdidas económicas. La brecha que existe aquí en We Techs radica en la falta de un enfoque sistemático para poder mejorar la calidad de los datos que se manejan en We Techs.
Al ser parte de equipo de analítica avanzada, la necesidad de trabajar con la mejor calidad de datos posible es una gran ayuda y reduciría los tiempos de creación de modelos predictivos, que es algo que la empresa actualmente esta dándole mucho enfoque es por eso que de ahí nace la oportunidad de la detección de anomalías en los datos antes de que estas lleguen a a la base de datos de la empresa. Pero primero se debe saber que nos permite mejorar la calidad de los datos de la empresa al detectar las anomalías y trabajarlas antes de ser utilizadas para los modelos de predicción. Esto nos permite lo siguiente:

- Datos limpios y mas precisos, debido a que la presencia de anomalías puede sesgar los modelos provocando predicciones inexactas, en donde limpiar la data no seria suficiente ya que la data venia desde un comienzo sucia y eso aun se puede traspasar a los datos pre-procesados.
- Se puede comprender mejor los posibles patrones que sigue la data, lo que permitiría ajustar mejor los modelos para poder capturar estos patrones y mejorar la precisión de los modelos.
- Se puede reducir el ruido en los datos, ya que la presencia de esto afecta al rendimiento del modelo. Eliminando o reduciendo este ruido podría ayudar al modelo a detectar patrones mas significativos.
- Se puede identificar problemas mas rápido lo que también reduce el tiempo de preprocesamiento de los datos, en donde también se simplifica este mismo paso debido a la reducción de anomalías.

Trabajando con los datos disponibles para la creación de modelos predictivos, se pudo notar que en varias fracciones de tiempo habían comportamientos muy extraños de los datos y no

se sabia porque los datos se comportaban de esa manera. Esto provoca incertidumbre a la hora de trabajar los datos, ya que no sabemos si fue un comportamiento provocado por la gente que trabaja en terreno, o si fue error de medición de la instrumentación entre otras cosas.

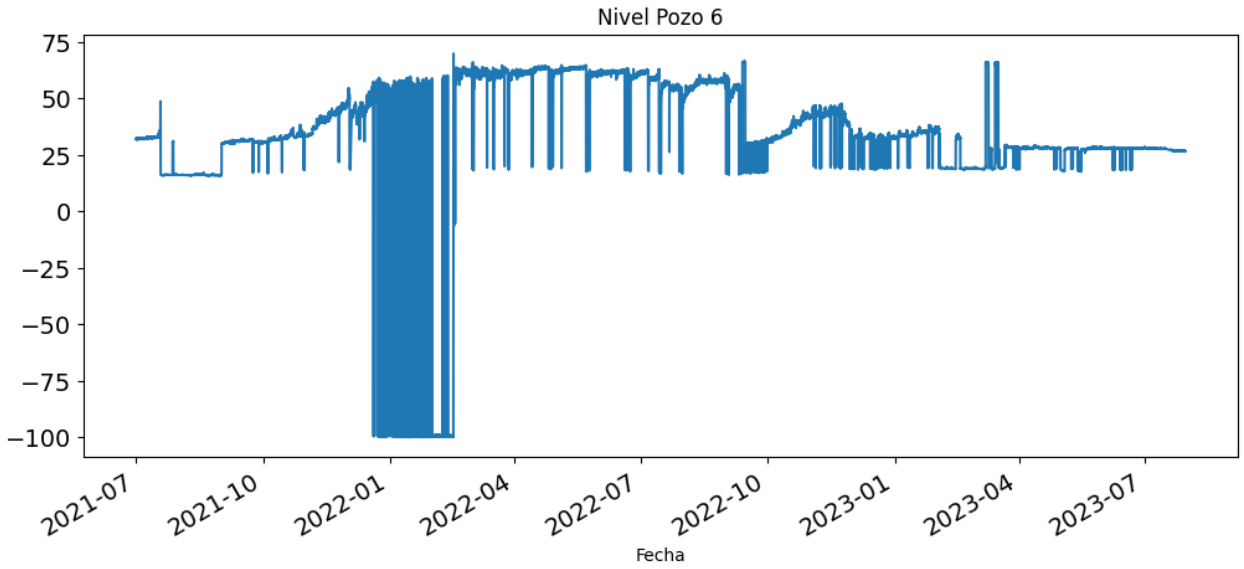

Figura 1: Pozo 6 Los Litres (Creación Propia)

Como se puede ver en la Figura 1 se tiene como ejemplo la serie de tiempo de el Pozo 6 de Los Litres, y como se puede observar entre Diciembre del año 2021 hasta Marzo del año 2022 tenemos un rango de datos que se comporta muy extraño, específicamente cuando los valores a los que llega son valores negativos y es imposible obtener valores negativos ya que los pozos no puede tener niveles negativos debido a que eso estaría diciéndonos que el agua se estaría saliéndose del pozo. Son cosas como esta que no se han ido detectando y tampoco documentado, lo que hace mas complicado el análisis de los datos, al no saber con certeza la naturaleza de los datos. Cabe destacar que la instrumentación

Es importante mencionar que se trabajara específicamente con la variable de nivel de los pozos, y la razón es bastante simple. Esto es porque el nivel es la única variable en los pozos que puede verse afectada naturalmente, ya sea por precipitaciones, temperaturas extremas, entre otras cosas. A pesar de que la detección de anomalías sera solamente aplicado a la variable de nivel de todas formas se utilizara la variable de caudal en el entrenamiento del

modelo ya que tiene directa relación con el nivel del pozo, debido a que el nivel del pozo disminuye, esto es porque el nivel se mide en metros, y es la distancia entre la superficie del pozo con la superficie del agua, por ende mientras mas lleno esta el pozo, menor es el valor del nivel (como se puede observar en la figura 2)

Figura 2: Nivel de un pozo (Creación Propia)

4. Objetivos generales y específicos

- **Objetivo General**: Mejorar la calidad del dato que se obtiene desde la base de datos de la empresa We Techs de los 13 pozos de El Soldado. (Ubicados en la Región de Valparaíso, en Los Litres (6 Pozos) y El Melón (7 Pozos)).

- **Objetivos específicos**:

  1. Mejorar las predicciones de las variables (Nivel, Volumen y Caudal) de los 13 pozos que se trabajan.
  2. Implementar el sistema de detección de anomalías mediante algoritmos de Machine Learning.

3. Documentar los datos anómalos identificados con fecha y su posible causa.

5. Estado del arte

La instrumentación utilizada en We Techs es fundamental para el proyecto ya que esta es la que se encarga de obtener los datos, la cual nos ofrece valiosa información, para poder observar patrones en los datos y poder también realizar predicciones sobre estos. Sin embargo, la presencia de anomalías en estos datos puede afectar en la interpretación de estos datos como también en la realización de modelos predictivos, por lo que la implementación de un modelo de detección de anomalías ayudaría a que estos sean precisos y que no se vean tan afectados por el estado de los datos.

Trabajando en la sección de datos de la empresa, al realizar las predicciones de los niveles de los pozos (ya que para la empresa en este momento es la variable más importante), se ha notado la falta de calidad de los datos, al igual que la falta de acceso a algún tipo de documentación del porque sucede este dato anómalo, esto provoca demoras al intentar buscar que fue lo que paso en cierta fecha, ya que puede ser que la razón de algún dato anómalo, sea por algún tipo de prueba de parte de la empresa que no fue notificada, como también puede ser un error de lectura, entonces tener documentado ayudaría mucho a los análisis de los datos y poder tomar decisiones informadas.

Es por eso que la detección de anomalías para la empresa We Techs no solo es un ejercicio académico, sino que también es un gran paso hacia la mejora de la calidad de los datos y que esto sea el primer paso para mejores practicas y avances de la detección de anomalías para la empresa.

5.1. Datos anómalos

En el entorno de la empresa de We Techs, la calidad del dato esta tomando cada vez mas importancia para poder realizar el trabajo con mayor exactitud y poder tomar decisiones informadas. Es por esto que bajo este contexto la detección de anomalías se presenta como un componente esencial para poder salvaguardar la calidad del dato, específicamente en los datos que se obtienen a través de la instrumentación, la cual nos proporciona con los datos de caudal, nivel y volumen de los pozos. También se debe aclarar a que se refiere con

dato anómalo. Los datos anómalos son observaciones en los datos, los cuales se desvían del comportamiento general del resto del conjunto de datos. Hay distintas causas que pueden formar un dato anómalo como por ejemplo:

1. **Valores extremos:** Son datos que se encuentran muy lejos de la mayoría de los otros puntos del conjunto de datos, estos pueden ser identificados con distintas medidas estadísticas como el rango intercuartil (IQR).

2. **Errores de entrada o medición**: Datos que son errores obvios, que están fuera del rango esperado, debido algún error en la instrumentación que lo mide.

3. **Inconsistencias lógicas:** Estos son datos que contradicen la lógica del problema, como por ejemplo bajo este contexto, los pozos no pueden tener nivel negativo ya que no tendría ningún tipo de sentido.

4. **Cambios bruscos:** También pueden notarse cambios bruscos en el comportamiento de los datos, lo que puede indicar un error o algún evento inusual.

5.2. Trabajos relacionados

El sistema de detección de anomalías en los datos a sido un tema para distintos investigadores en distintos rubros. Primero tenemos a Crépey y su equipo realizo una investigación y desarrollo sobre un sistema de detección de anomalías en series de tiempo financieras, a través de redes neuronales y análisis de *componentes principales (PCA)*[2], en donde utilizan PCA como extractor de características en los datos, para luego aplicar las redes neuronales para la detección de anomalías, lo cual no separan en dos partes. La primera parte es identificar series temporales con anomalías evaluando la propensión de la serie temporal a estar contaminada, esto lo ven reflejado en el puntaje de anomalía, para luego calibrar una *red neuronal feedforward*[3]. La segunda parte localiza la anomalía entre los valores observados de las series temporales identificadas como anómalas (Crépey et al., 2022).
Yanjun Zhou, también con un equipo, se inspiraron en el concepto de *granularidad*[4] de la

[2]El Análisis de Componentes Principales (PCA) es una herramienta que ayuda a simplificar y entender conjuntos de datos complejos. Básicamente, encuentra patrones importantes en los datos y los resume de manera que podamos entenderlos mejor.

[3]Es una red neuronal simplificada en donde la información fluye en una sola dirección.

[4]Se refiere al nivel de detalle o a la complejidad de la información utilizada para entrenar o evaluar un modelo.

información aplicado al proceso de modelado del sistema. Ellos proponen un *modelo Markov granular*[5] para la detección de anomalías, lo cual utilizan los cambios de amplitud y forma de los datos de la serie temporal, de esta forma pueden representar la granularidad de la información del intervalo, para luego utilizar el algoritmo de agrupamiento *Fuzzy C-Means (FCM)*[6], para generar diferentes estados de granularidad de información (Zhou et al., 2022). Nikolay Laptev propone un marco genérico y escalable para la detección automatizada de anomalías. Utiliza un sistema en Yahoo, llamado *EGADS*[7] (Extensible Generic Anomaly Detection System), el cual utiliza una colección de modelos de detección de anomalías y pronostico con una capa de filtrado de anomalías para lograr un detección precisa y escalable Laptev et al., 2015. Alexander Lavin y su equipo proponen *"Numenta Anomaly Benchmark"*[8] (NAB), el cual intenta proporcionar un entorno controlado y repetible de herramientas de código abierto para probar y medir algoritmos de detección de anomalías en datos en tiempo real. NAB evalúa los detectores en un conjunto de datos de referencia con datos de series temporales del mundo real etiquetados (Lavin y Ahmad, 2015).

5.3. Métodos mas comunes

- **ARIMA:** Este algoritmo se utiliza si es que nuestra data muestra una clara tendencia den estacionalidad.
- **Isolation forest:** Este es un algoritmo de machine learning para la detección de anomalías. Esta basado en el concepto de que las anomalías son puntos en la data que son pocos y diferentes comparado con el resto, lo cual haría que fueran mas fáciles de aislar que el resto.
- **LSTM Autoencoder:** Si la data tiene patrones complejos que métodos tradicionales

[5]En el contexto de granularidad el modelo markov se refiere a el nivel de detalle con la que se discretiza o se divide el espacio de estados del sistema modelado. El espacio de estados se refiere a todas las posibles condiciones que puede tomar un sistema.

[6]Algoritmo de agrupamiento que permite que los puntos de datos pertenezcan a múltiples grupos con diferentes grados de membresía. Esto permite reflejar la idea de que los puntos pueden tener asociaciones parciales con varios grupos.

[7]El primer sistema integral de deteccion de anomalias que es flexible, preciso, escalable y extensible.

[8]Modelo que intenta proporcionar un entorno controlado y repetible de herramientas de código abierto para probar y medir algoritmos de detección de anomalías en datos en tiempo real.

no pueden capturar, es bueno considerar este método con LSTM autoencoder. Es un tipo de red neuronal que aprende a reconstruir la data, y las anomalías pueden ser detectadas como data points donde la red neuronal tiene un alto grado de error al reconstruirlo.

- **Local Outlier Factor (LOF):** Este método calcula la desviación de densidad local de un data point dado comparado con sus vecinos. Considera como outliers las muestras que tienen substancialmente menor densidad que sus vecinos.
- **One-Class SVM:** Este metodo se usa para *"Novelty detection"*, que son las propiedades de la data que son aprendidas con One-Class SVM, y luego las anomalias pueden ser detectadas como data points que no entran en las propiedades aprendidas.

Existe también una librería muy potente dentro de Python, la cual es llamada Darts, la cual es una librería que se usa específicamente para trabajar con series de tiempo, ya sea haciendo forecasting, identificando patrones, interpolación de datos faltantes y mas. Esta herramienta es la que se empezara a utilizar en la empresa We Techs para las predicciones de forecasting de los datos. Esto seria algo positivo no solo para la realización del proyecto si no que para la empresa en general ya que Darts incluye herramientas para la creación de modelos de detección de anomalías en los datos, las cuales no serán utilizadas en este momento ya que aun se esta observando el alcance de la librería en lo que viene siendo timeseries forecasting. Dado la naturaleza de los datos en We Techs, hay dos opciones que son las que mejor trabajan los datos anómalos, estos son *K-Means Scorer* y *Filtering anomaly model*.

6. Solución escogida

La solución la cual se definió para realizar en la empresa es, utilizar modelos de detección de anomalías al momento de obtener los datos a la base de datos de We Techs, de esta manera podemos saber que datos son anómalos en los datos que se utilizan para poder hacer predicciones de las variables de los pozos. En un principio se identificaron dos soluciones contundentes para poder llevar a cabo los objetivos. El primero fue el mas claro y este es la utilización de machine learning para entrenar un modelo de detección de anomalías, y luego implementarlo a la infraestructura de la empresa. Y la segunda posible solución fue la idea de implementar inteligencia artificial, debido a su rápido crecimiento y cada vez mas importancia en el mundo laboral en general, la idea de utilizar IA para que detecte anomalías no era un mala opción. Se termino inclinando por machine learning debido a que la empresa aun no esta trabajando con IA, por lo que de prefirió partir de cosas mas simples, en caso de errores u otros problemas.

La idea del modelo es implementarlo en la infraestructura AWS de la empresa en donde como se menciono anteriormente la idea es que el modelo sea implementado antes de guardado en la base de datos de la empresa, si no que sea justo después de la transformación de los datos. En la figura 3 se puede ver donde se quiere implementar el modelo para que se entiende mejor.

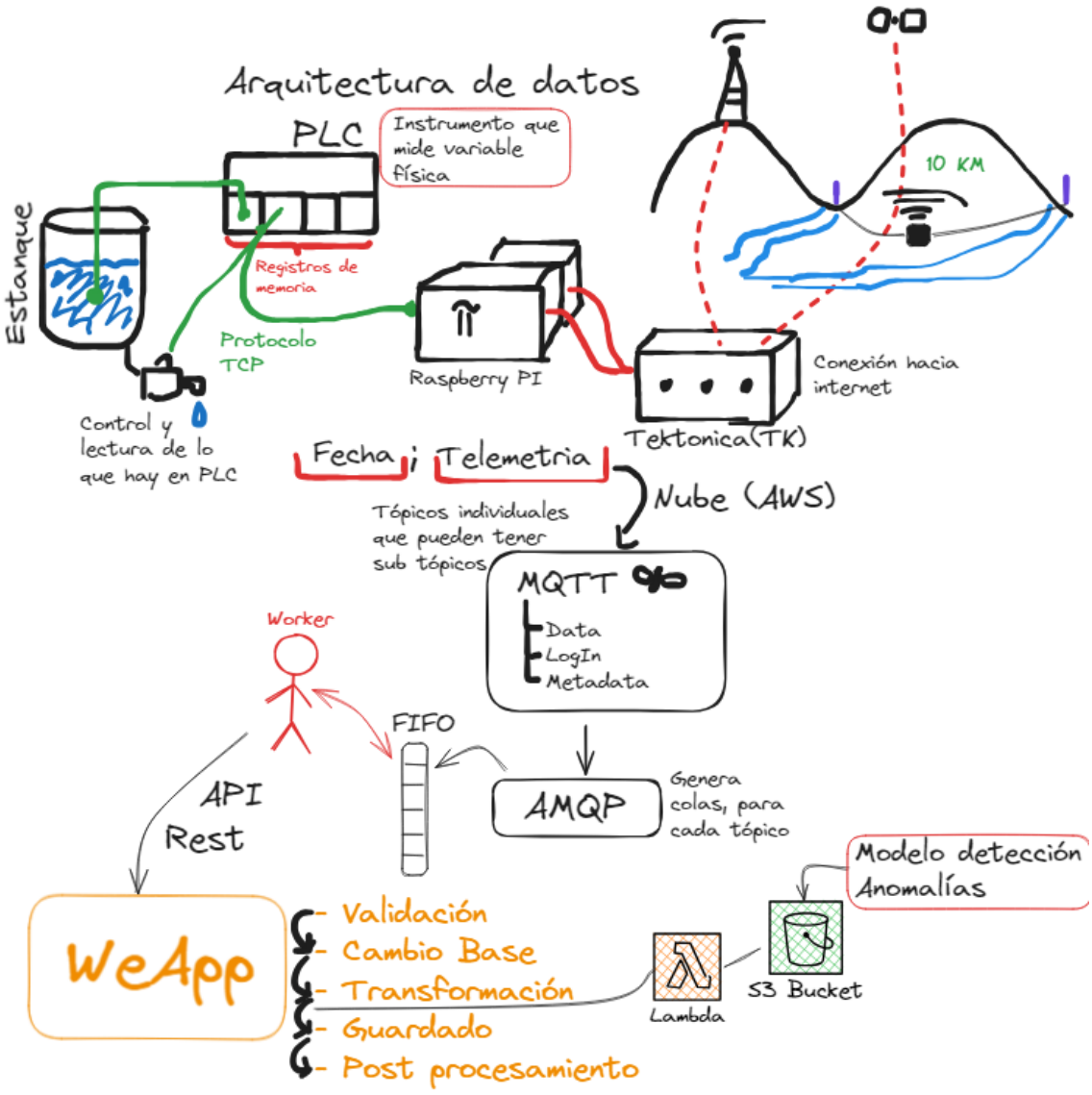

Figura 3: Arquitectura de los datos de We Techs (Creación Propia)

6.1. Evaluación de necesidades

Primero se buscaron las necesidades de la empresa, en donde se dio a conocer la absoluta necesidad de empezar a mejorar la calidad del dato dentro de la empresa. Con esto se identifico la oportunidad junto a empleados de la empresa que se necesita la detección de datos anómalos en la data ya que no se tiene como saber de su presencia hasta el momento de trabajar con ellos, lo que ralentiza las predicciones y el análisis de estas para el equipo de analítica de data.

6.2. Camino a la solución

Se dejo en claro que se necesitaba un sistema de detección de anomalías a través de Machine Learning, lo que llegamos a lo que se podría llamar "sub soluciones", ya que se tenia que seleccionar un modelo de machine learning para poder tener el modelo de detección de anomalías. Por lo que se ideo el plan de partir utilizando modelos mas simples y poder observar como estos modelos se comportaban y que tan buenos eran en detectar anomalías. Estos modelos están dentro de los modelos no supervisados, esto se hizo debido a que no se contaba con etiquetas para poder utilizar como validación en modelos supervisados. Se hicieron modelos de *isolation forest*[9], *local outlier factor*[10] y *STL loess*[11]. Los primeros dos que se mencionaron son modelos no supervisados y los mas comunes y efectivos en detectar anomalías en los datos, con ese criterio se decidió seleccionarlos como los principales modelos en probar.

Haciendo el análisis de las variables que se trabajan en los pozos de El Soldado, las cuales son Caudal, Volumen y Nivel, pudimos observar que la única variable que tiene variaciones naturales (como por ejemplo precipitaciones, temperatura entre otras cosas) es el Nivel, ya que el caudal es manipulado por la gente que trabaja, mientras que el volumen es el caudal por el tiempo prendido. Es por eso que se concentro mas que nada en el nivel para poder hacer el modelo.

[9] Algoritmo de Machine Learning el cual busca datos inusuales en un grupo de datos de manera rápida.

[10] Algoritmo de Machine Learning para identificar datos anómalos, el cual evalúa la densidad relativa de las instancias en comparacion con sus vecinos locales.

[11] Seasonal Trend decomposition es una técnica de estadística y análisis de series temporales para descomponer la serie temporal en tres componentes, la cual se puede utilizar para la detección de anomalías en la componente residual después de la descomposición.

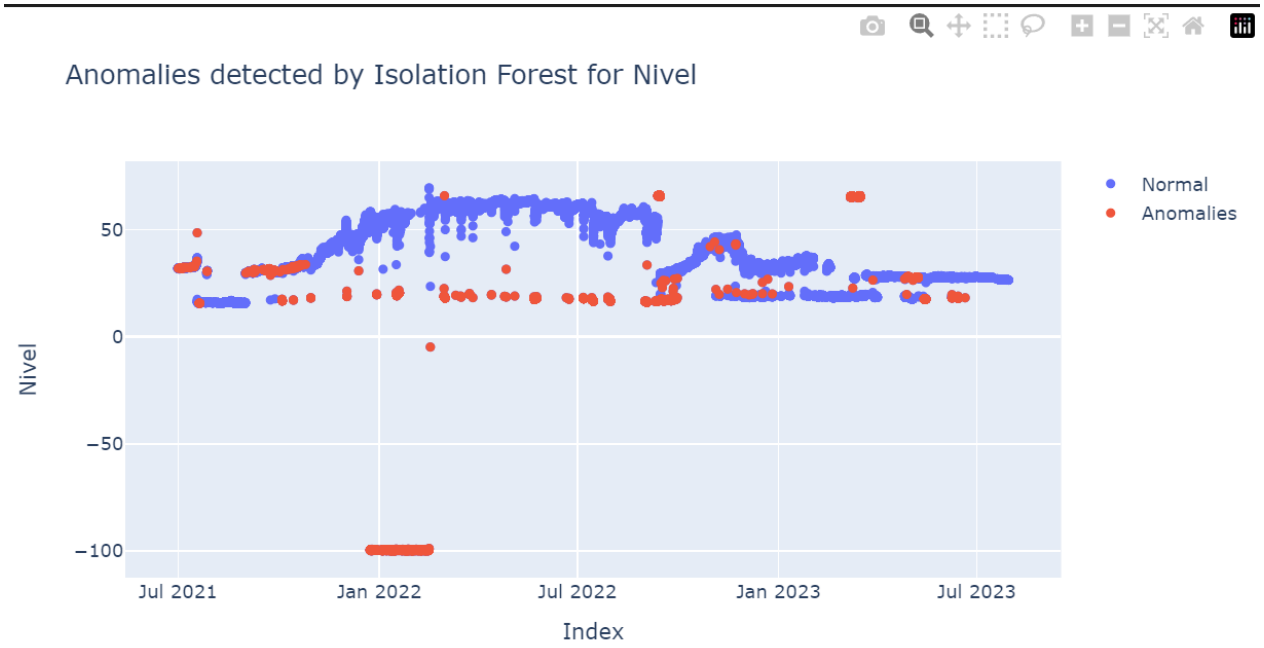

Figura 4: Isolation Forest (Creación Propia)

En la Figura 4 podemos ver el resultado del modelo de Isolation forest, en donde podemos ver que para ser un modelo no supervisado marco bastante bien los datos anómalos, como los datos negativos que tenemos en nivel, lo cual claramente no tienen sentido debido a que nos estaría diciendo que el nivel del agua esta sobrepasando el limite del pozo ya que el nivel se mide desde la superficie del agua hasta la superficie del pozo en lugar del fondo del pozo. Pero al mismo tiempo como podemos ver hay puntos que se consideran anómalos que la verdad no tienen mucho sentido que se consideren anómalos.

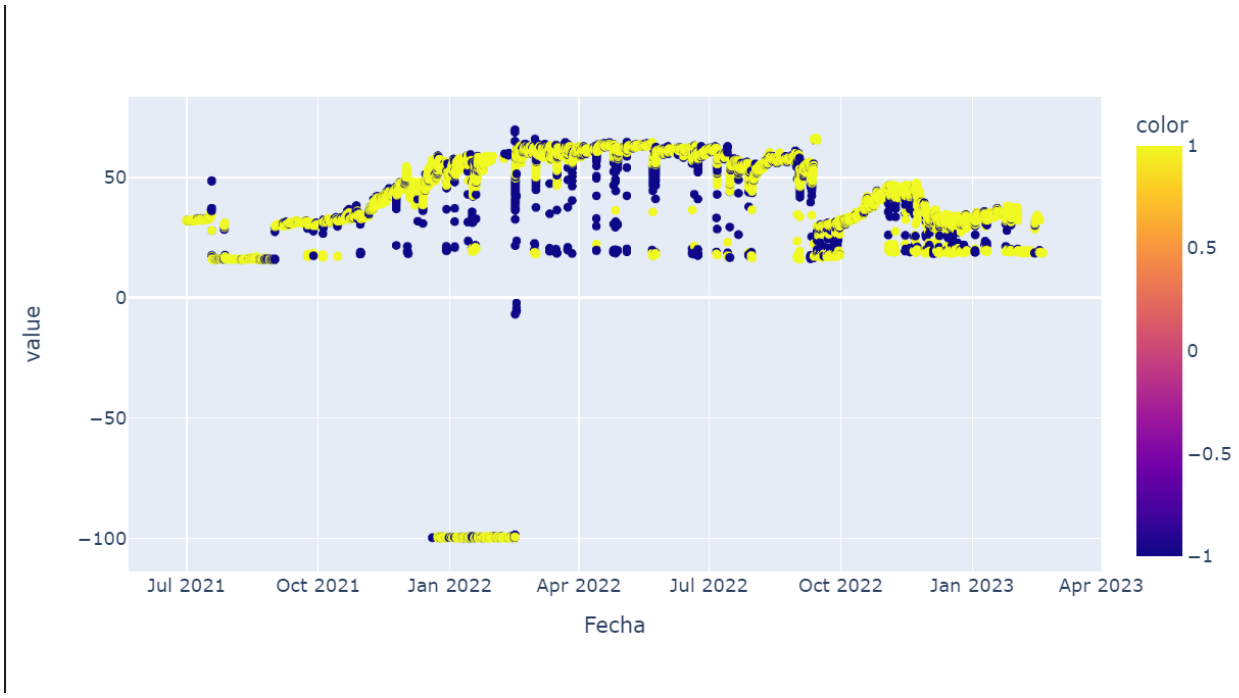

Figura 5: Local Outlier Factor (Creación Propia)

Por otro lado en esta imagen tenemos la detección de anomalías realizada con LOF, y como podemos ver simplemente no funciono para nada bien, ya que como se puede ver lo que esta marcando como anomalía se puede ver que es muy al azar (1 es dato normal y -1 es anómalo), y como se puede ver este marco los datos negativos como datos normales, por lo que no se siguió experimentando con este modelo.

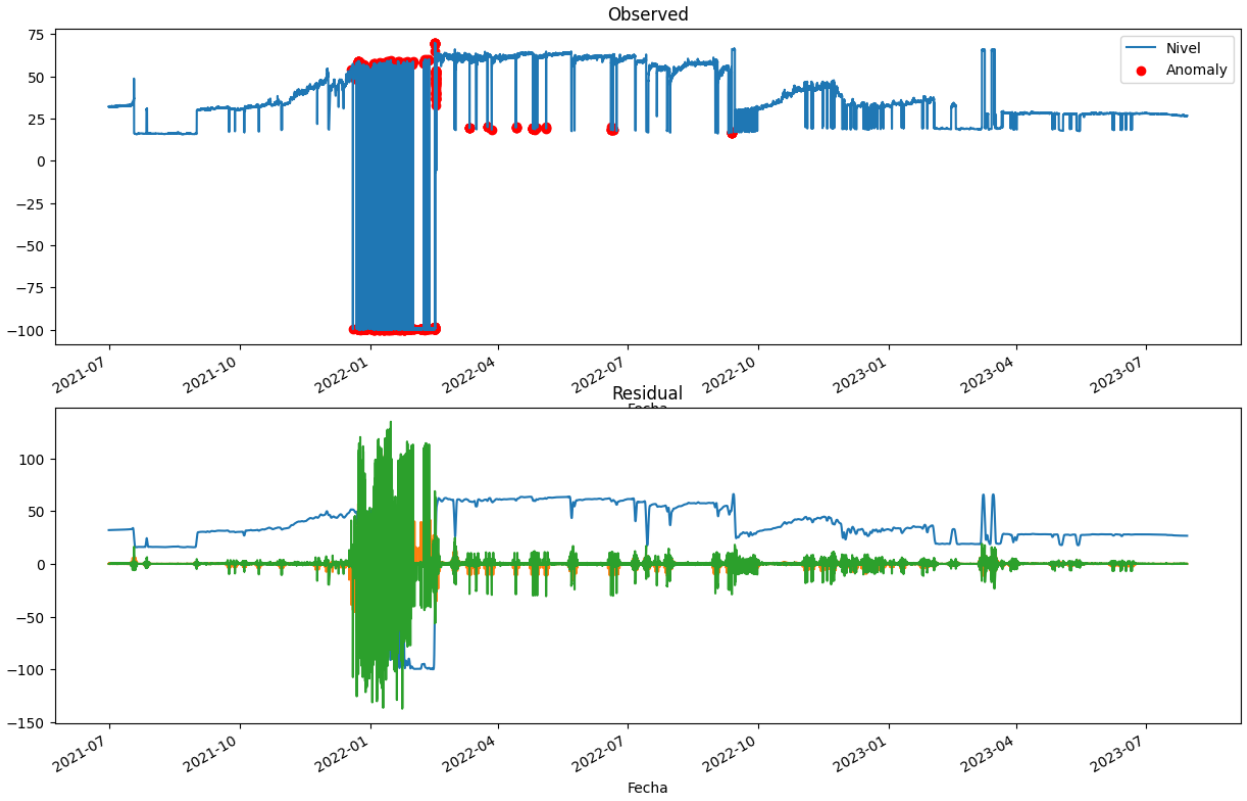

Figura 6: STL anomaly (Creación Propia)

Luego tenemos la detección de anomalías realizada con STL. En donde al igual que Isolation forest dio resultados bastante sorprendentes y acertados en el ámbito de mostrar que datos son anómalos y que datos no lo son, pero a diferencia de isolation forest no hay tantos puntos marcados a lo largo de la data que pareciera que es normal, por lo que se puede concluir que de los 3 modelos no supervisados utilizados y optimizados es STL.

Luego se quiso utilizar algo mas robusto, pero aun relativamente simple, por lo que se decidió utilizar LSTM Autoencoder, pero de manera no supervisada ya que la empresa no cuenta con etiquetas para saber que es algo anómalo y que no. Se intento generar las propias etiquetas, pero como se menciono anteriormente no se tiene una idea de que es anómalo y que no, solamente se sabe que los datos no pueden ser negativos.

Se entreno un modelo por cada sector de El Soldado, ya sea en Los Litres o El Melón. Esto se debe a que la empresa quería la menor cantidad de modelos posibles. Debido a que los pozos de los litres tienen un comportamiento parecido aunque no igual; lo mismo pasa con

los pozos de El Melón, por lo que se entreno un modelo por cada uno.

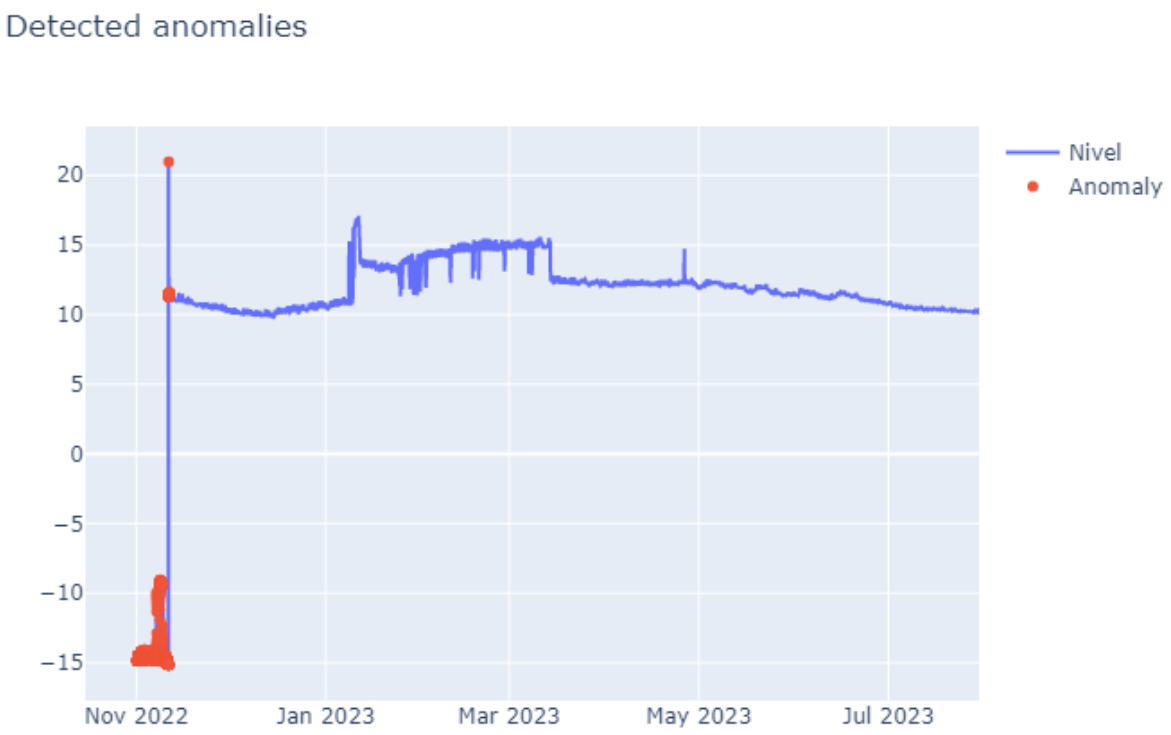

Figura 7: LSTM Autoencoder Pozo 3 El Melón (Creación Propia)

Se utiliza el ejemplo de la Figura 7, ya que el gráfico incluye valores negativos, y como se puede observar todos esos valores se están considerando como anomalías, entre otros puntos que a simple vista se puede considerar anómalos.
Luego de todo esto se hizo un análisis mas a la variable de nivel y se puede ver que hay dos tipos, Nivel estático que se refiere a cuando el caudal es igual a 0, por ende este es uno de los mas importante ya que es el único que el hombre no puede interferir en como se comporta y por otro lado tenemos el nivel dinámico, que aquel cuando el caudal es mayor que 0.

Luego de las pruebas con todos los modelos mencionados anteriormente se trabajo en un

prototipo de las alertas, simplemente para observar como se verían. Esto se probo de manera local por lo que el código no seria igual y se tendría que utilizar SES que es aquella función que funciona en AWS para enviar correos.

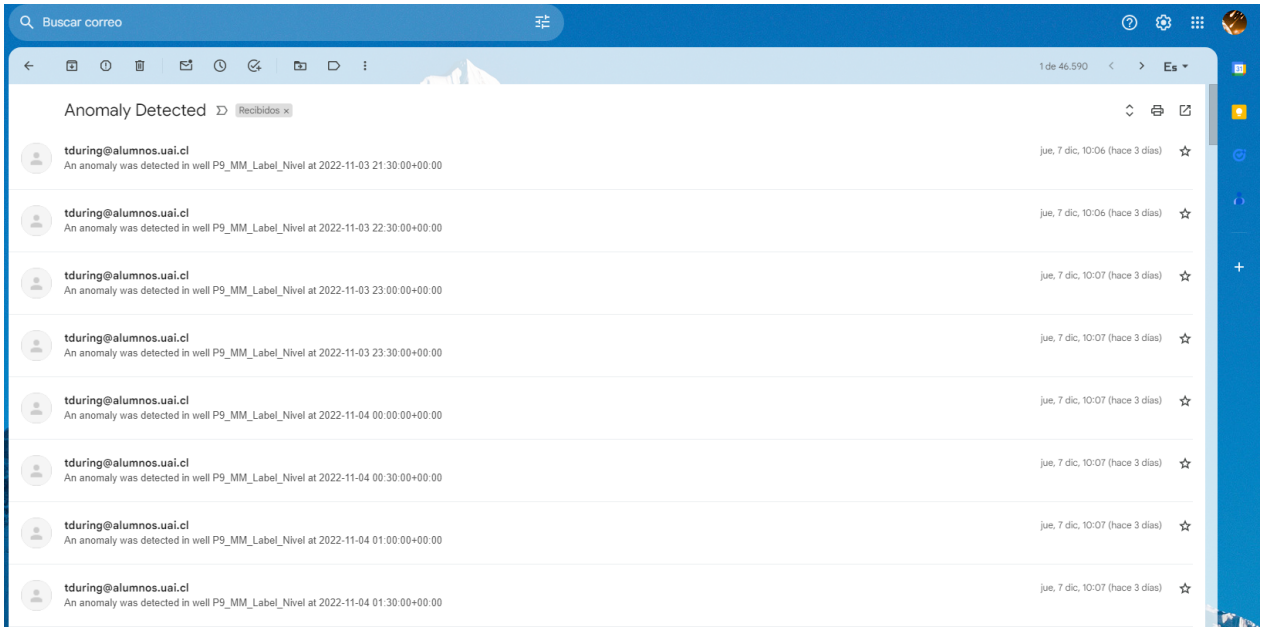

Figura 8: Alerta de anomalías (Creación Propia)

Como se puede observar en la figura 8 se muestra un simple correo en donde se mandan todas las anomalías detectadas de un pozo en especifico.

6.3. Plan de implementación

Para llegar a la fase final que es la implementación del proyecto primero se tiene que pasar por las primeras fases que se definieron, este plan consta de 3 fases a lo largo de 5 meses.

6.3.1. Fase de preparación

1. **Definición de requisitos y objetivos:** Se deben establecer claramente los objetivos del proyecto con la implementación del modelo de detección de anomalías, esto se puede ver en los objetivos planteados anteriormente.

2. **Acceso y preparación de datos:** Necesitamos conseguir el acceso a los datos brutos de la empresa. Al comienzo se quería observar los datos anómalos en la data bruta de

We Techs por lo que no se necesitaba realizar pre procesamiento de datos. Por lo que lo único que se realizo para esto fue separar los .csv que se obtuvieron de acuerdo al pozo que se esta hablando, lo que quiere decir que obtuvimos 13 .csv en total, lo que hace que tengamos que probar los modelos en todos, pero esto se hace simplemente cambiando el .csv llamado ya que todos tienen el mismo formato y nombre de columnas (Fecha, Volumen, Caudal y Nivel).

Luego de esto se pudo identificar la presencia de datos repetidos en la data, lo cual provocaba ruido en la data, este ruido hace que el modelo tenga un menor rendimiento ya que capta patrones que no deberían estar, estos datos repetidos eran provocados por la *query*[12] que se realizo, por lo que posteriormente se decidió eliminar los datos repetidos y simplemente quedarse con 1 para cada dato repetido.

6.3.2. Fase de desarrollo y evaluación

1. **Exploración y selección de modelos:** Queremos probar distintos modelos de detección de anomalías, y probar su eficiencia y rendimiento para poder saber si el modelo realizado sirve para nuestra data.

2. **Optimización del modelo seleccionado:** Seleccionar el modelo mas apto y realizar los necesarios ajustes y optimizaciones para mejorar su rendimiento.

   Por otro lado se entreno otro modelo sin los datos repetidos mencionados anteriormente para poder hacer comparaciones, en donde no hubieron diferencias fuera de 1 o 2 puntos menos en el entrenamiento de datos no repetidos.

6.3.3. Fase de implementación y monitoreo

1. **Despliegue del modelo:** Una vez optimizado el modelo este se preparara para su implementación

2. **Integración con la infraestructura empresarial:** Asegurar una integración fluida del modelo a la infraestructura existente en We Techs, verificando compatibilidad, escalabilidad y seguridad del despliegue.

[12]Solicitud o petición de información a la base de datos.

6.3.4. Resultados esperados

- Por el lado de mejorar las predicciones de forecasting que se están realizando en la empresa, se espera que el modelo luego de 1 año los datos del modelo se note un cambio en la calidad de estos. No se espera un cambio perfecto pero si que haya un cambio notorio en la calidad de los datos, como en tendencias, coherencia de los datos entre otras cosas.

- En la parte de implementación del modelo de detección de anomalías, se espera que el modelo pueda detectar una buena cantidad de los datos, alrededor del 75 % de los datos que se marquen como anomalías sean correctas (que no sean falsos positivos o falsas anomalías), ya que como se ha mencionado anteriormente es muy difícil hacer que el modelo sea perfecto especialmente la primera iteración de este.

- Finalmente en la documentación de las anomalías se espera que haya una persona o grupo que al detectarse una anomalía se encargue de observar si hubo algún problema. La alerta documentara la fecha y el pozo, pero no tendrá la capacidad de saber la causa del porque se detecto una anomalía, por lo que se necesitaría a alguien encargado de esa parte y poder documentarlo manualmente.

7. Evaluación de riesgo

Para evaluar los riesgos que se pueden presentar en el desarrollo de este proyecto se utilizo la matriz de riesgo para poder observar posibles acontecimientos, y ver que tan probables son que pasen y al mismo tiempo que tan grave seria si es que estos llegan a pasar.

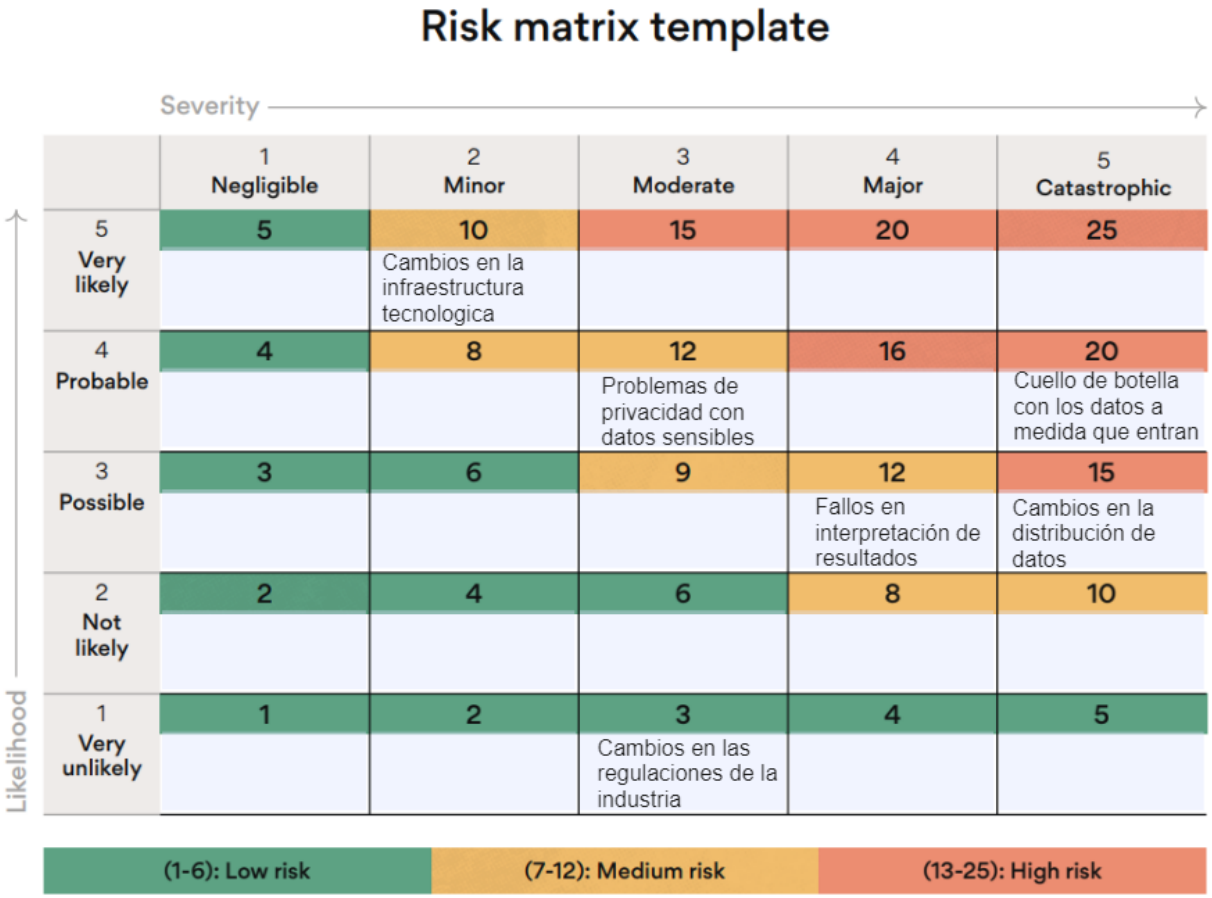

Figura 9: Matriz de Riesgo (Creación Propia)

1. **Cambios en la infraestructura tecnológica:** Este punto esta en muy probable ya que es prácticamente un hecho que esto pasara en la empresa, es por eso que esta puesto en el nivel 5 de probabilidad. Pero esto se pone en severidad nivel 2 ya que se dio a conocer que lo que se cambiara por ahora de la infraestructura es simplemente un

cambio de *monolito*[13] a *micro servicios*[14].

El proyecto para este cambio de infraestructura lo que quiere hacer es cambiar la infraestructura digital sobre la que esta montada la WeApp (aplicación con la que trabaja We Techs) y todos los otros servicios, esto hará que pase de ser un monolito a micro servicios. Esto quiere decir que siendo un monolito, significa que esta construida como una única unidad grande e indivisible, por lo que el proyecto de infraestructura busca dividir la aplicación en piezas mas pequeñas e independientes, cada una encargada de una función especifica (micro servicios). Esto permite mayor flexibilidad, escalabilidad y mantenimiento mas sencillo en comparación al monolito.

Mitigación: Ya es un hecho el cambio de infraestructura, por ende el plan seria aprender lo mas posible de la nueva infraestructura y poder adaptar el modelo para tenerlo preparado si es que es necesario. Por el plan que tiene la empresa los modelos deberían funcionar en ambas infraestructuras.

2. **Problemas en la privacidad con datos sensibles:** Esto puede provocar limitaciones a el acceso de los datos etiquetados para el entrenamiento que pueden contener información sensible. Como la necesidad de requisitos de consentimientos para la utilización de datos. Aquí en We Techs lo malo es que si han habido problemas con la privacidad de los datos, no se ha sabido respecto a ello ya que no hay control de usuario ni políticas al respecto.

   Esta puesto en probable, debido a que la empresa no tiene como saber si hubo alguna falla con la privacidad de los datos, lo que nos quiere decir que no tienen una forma de saber si perdieron datos sensibles o si se hicieron públicos. Pero por otro lado se puso en moderado porque aunque probablemente han habido problemas de este tipo no se ha visto un efecto tan grande como para que sea notorio, pero por otro lado generalmente la perdida de privacidad de los datos sensibles es algo muy peligroso y es por eso que se dejo en un punto medio en el nivel 3.

   Mitigación: Establecer procesos de análisis de fallos pasados, para poder mejorar el modelo y los procesos de interpretación. También se puede aplicar capacitación de los

[13] Arquitectura monolítica es donde toda la aplicación se desarrolla como una única unidad, ya sea la interfaz lógica de negocios, etc.

[14] Arquitectura de micro servicios es cuando la aplicación se divide en servicios mas pequeños e independientes, donde cada uno tiene su propia lógica de negocios y base de datos.

empleados.

3. **Fallos en la interpretación de resultados:** No podemos obviar el error humano a la hora de observar los resultados, siempre habrán errores a la hora de ver resultados lo que lamentablemente provocaría impactos significativos a la hora de analizar los resultados, lo que dificultaría observar bien que datos son anómalos y cual es la tendencia del modelo al detectar estas anomalías.

   Es por eso que esta puesto en el nivel 3 de probabilidad ya que como se menciono anteriormente el error humano es algo que siempre esta presente. Y luego en severidad nivel 4 o mayor debido a que una continua falla en la interpretación de los datos puede hacer que el modelo a la larga no sirva para nada.

   Mitigación: Establecer procesos de análisis de fallos pasados, para poder mejorar el modelo y los procesos de interpretación. También se puede aplicar capacitación de los empleado

4. **Cambios en la distribución de datos:** que suceda esto es posible y es algo que no se puede descartar ya que es algo que pide el cliente. Actualmente la data esta cada 5 minutos, pero si el cliente necesita la data cada 1 minuto se tendrá que ajustar la instrumentación para obtener los datos cada 1 minuto, y esto haría que tengamos una cantidad bastante mayor de datos lo que podría provocar que el modelo de detección de anomalías puede que falle debido a las grandes cantidades de datos ya que estaría afectando a la capacidad de computo de la infraestructura.

   Hablando de la probabilidad se uso en el nivel 3 de probabilidad ya que la distribución de los datos depende del cliente y como quieren ellos la distribución de los datos. Actualmente los datos están cada 5 minutos, pero eso no quiere decir que siempre se quedaran de esa manera. Por otro lado la severidad se puso en el nivel 5, por que es subjetivo a cuanto cambie la distribución de los datos, ya que cambiar de 5 a 10 minutos quizás no afecte en tanto al modelo, pero si se cambia a 1 hora por ejemplo quizás el modelo no funcione correctamente y falle un poco a la hora de detectar las anomalías.

   Mitigación: Plan de mitigación seria, aplicar técnicas de adaptación de dominio, para que de esta manera el modelo no olvida patrones aprendidos previamente, y luego simplemente se podría cambiar un poco para adaptarse a la nueva frecuencia de datos.

5. **Cambios en las regulaciones de la industria:** Este punto es algo que es muy difícil que suceda a tal magnitud que afecte. Pero en el caso de que hallan cambios grandes en la industria, pueden haber distintas cosas que podrían afectar, como por ejemplo, cambios en la retención de datos, queriendo decir que se establezcan limites en la retención de datos lo que afectaría a la disponibilidad de datos históricos. Esta puesto como baja probabilidad ya que no se ha sabido aun cambios en la industria (minera y de energía), por un lado la minera ya hizo los cambios para este año y la de energía no se han encontrado planes para realizar cambios en esta.

   Se puso este riesgo en nivel 1 de probabilidad ya que se hizo una investigación sobre las dos principales industrias en las cuales esta presente We Techs, en donde la industria minera, los cambios que tenían planeado para 2023 ya fueron realizados por lo que no debería haber un cambio tan cercano, mientras que la severidad de este es moderado, ya que pueden incluir regulaciones que afecten a la empresa y la obtención de los datos.

   Mitigación: Lo mas simple como plan para este suceso es estar informado sobre los posibles cambios en la industria y como esto podría afectar a la empresa. También se puede anteponer a esta posibilidad y tener un modelo bastante flexible.

6. **Generación de cuello de botella con los datos:** Implementar la detección de anomalías justo antes de incorporarlas a la WeApp puede que provoque un cuello de botella debido a la cantidad de datos que pasan, puede que el modelo de detección se demore en identificar un dato y esto provoque el cuello de botella, ralentizando mucho la obtención de datos.

   Esto se puso como nivel 4 de probabilidad debido a que depende mucho de la velocidad del modelo para detectar anomalías, y al ser la primera iteración de modelo quizás no sea tan efectivo, por lo que se considera probable que se genere un cuello de botella, lo que seria severidad nivel 5 ya que al generarse un cuello de botella los datos no llegan a la WeApp cuando deberían llegar y podrían perderse muchos datos.

   Mitigación: Esto se puede mitigar incluyendo una nube entre medio, que a medida que vayan saliendo los datos, en lugar de ir directamente a la WeApp entren a la nube donde son analizados y así los datos no dejan de llegar y no se pierden datos.

8. Evaluación económica

Se hizo una breve investigación de los costos asociados a lo que es la detección de anomalías y todo lo que conlleva. El modelo de de detección de anomalías y su aplicación en si no tiene ningún costo evidente fuera de las horas hombre o el uso de recursos en el PC, como también el uso de electricidad. Por lo que se acudió a cosas que la empresa ya tenia pero que son necesarias para poder aplicar el modelo. Estas cosas son:

- **Costos de instrumentación**, es lo primero que se considero. Queremos saber los costos de la implementación de los medidores de nivel y caudalimetros que son los que están relacionados con los valores que queremos detectarles valores anómalos. Lamentablemente no se ha podido conseguir el dato de esto.

- **Costos de transporte hacia terreno**, ya que muchas veces para revisar la instrumentación hay que hacer salidas a terreno, y claramente el viaje hacia allá tiene un costo asociado, el cual aun no se ha proporcionado.

- **Costo de capacitaciones**, aquí la empresa contrato varias asesorías o capacitaciones para distintos sectores de TI en la empresa. Pero los dos que mas nos interesan son las asesoría para la célula de data que la que se encarga de realizar las predicciones y los análisis de datos, la cual se contrato a la empresa MyFutureAI y su costo asociado fue $3.500.000 CLP por 32 horas de trabajo, lo cual se traduce a $109.000 CLP por hora, en donde se asignan 3 personas que pueden realizar preguntas y que tienen que ser parte de las reuniones que se hacen, pero de todas formas pueden haber cualquier cantidad de personas extra en las capacitaciones, pero no pueden interferir en ellas, por lo que sale bastante a cuenta para la empresa si es que se mete mas gente y no solo las 3 personas designadas.

- **Costos de tiempo y personal**, tal como dice el nombre y como se menciono anteriormente existe el costo de horas hombre para realizar el proyecto como también mantenerlo.

- **Costos de mantenimiento de infraestructura**, este es uno de los mas importante que se mantenga ya que sin la infraestructura no podemos obtener los datos que

necesitamos para poder realizar los modelos. Este mantenimiento cuesta $4.500 USD mensuales, lo que equivale a $3.966.030 CLP mensuales.

Como se puede ver se tienen en cuenta todos los gastos que han habido para poder tener acceso a los datos y que funcione bien.
Por otro lado si hablamos de ganancias, por ahora se puede decir que se ganaría tiempo a la hora de hacer las predicciones ya que gran parte de los datos anómalos serán trabajados antes porque serán avisados de la presencia de estos, también mejorando la calidad del dato se mejoraran las predicciones y por ende se tendrán clientes mas felices que es otra ganancia no monetaria. Esto también puede llevar a la obtención de mas clientes o la fidelización del cliente. No se tiene una forma clara de ganancias monetarias, por lo menos en el corto plazo ya que el modelo se debe dejar corriendo por harto tiempo para poder observar mas resultados y ver de mejor manera el comportamiento del modelo.

8.1. Análisis de sensibilidad

Para la realización del análisis de sensibilidad no se tienen datos numéricos claros por lo que este análisis se enfocara en distintos aspectos en donde se podría aplicar este tipo de análisis sin necesariamente incluir datos numéricos.

- **Sensibilidad en los datos de entrada:** Se debería evaluar como el cambio en los datos de entrada respecto a la temporalidad de los datos afectaría en el modelo. Esto solamente se podría hacer a través de simulaciones y ver como afectaría al modelo de detección que los datos cambien su temporalidad de 5 minutos a otro valor.
- **Sensibilidad en la elección del algoritmo de predicción:** A pesar de que ya se realizo como cambia la precisión de la detección de las anomalías aun se podría utilizar mas modelos y ver como cambia la detección dependiendo del modelo.
- **Sensibilidad selección de características:** Analizar como la inclusión o exclusión de ciertas variables impacta en la capacidad del algoritmo para detectar anomalías. En este caso no existen muchas variables dentro del dataframe que se esta utilizando. Con la empresa se ha hablado en poder incluir otras variables que podrían ayudar al modelo

a detectar mas patrones y mejorar la detección de las anomalías, pero por ahora no se tiene nada claro.

- **Sensibilidad en los umbrales de detección:** Se puede evaluar como variar los umbrales de tiempo para la detección de anomalías afecta en la cantidad y precisión del modelo. Esto refiere al momento de realizar el modelo se debe elegir un umbral de tiempo en donde se divide el entrenamiento y prueba del modelo. Actualmente se tiene una división como del 70 % para el entrenamiento y 30 %. Se fue probando con cantidades como un 80 %-20 % pero el modelo no funcionaba, y si se lograba ejecutar no se realizaba bien la detección, ya que marcaban datos muy al azar. Pero actualmente con el 70 %-30 % funciona bastante bien.

9. Metodología

- **Mejorar las predicciones del Nivel de los pozos:** Primero se tomara en cuenta la forma actual de como se realizan las predicciones con los datos actuales presentes en la empresa, y ver los resultados que se obtienen. Luego, con esto se puede obtener la precisión de los modelos realizados actualmente y así poder guardar estos valores.
  Luego de la implementación del modelo se tendrá que esperar un tiempo de 1 año largo antes de que se note algún cambio en los modelos significativos ya que la cantidad de datos buenos sera considerable luego de ese tiempo. Una vez se tenga la cantidad de datos necesario es momento de aplicar un modelo predictivo con la nueva data y luego comparar los resultados y precisiones de los modelos anteriores.

- **Implementar un sistema de detección de anomalías:** Para esto se debe primero entender la infraestructura de la empresa (en este caso se utiliza AWS), e identificar como poder guardar el modelo y que se pueda utilizar (se utiliza un bucket S3). Luego se tiene que tener en cuenta los costos adicionales que viene sujeto a la implementación. Con el modelo ya implementado se puede entrenar este mismo desde la misma infraestructura de AWS lo que ahorra mucho tiempo a la hora de ir actualizando el modelo debido a la data nueva.

- **Documentación de datos anómalos identificados:** La realización de este punto,

depende si el punto anterior fue completado, ya que se necesita que el modelo este funcionando para que las alertas y posterior documentación pueda funcionar. Por ende luego de que el modelo funcione, se debe realizar una alerta que avise cuando hay una anomalía, de esta manera luego al revisar las anomalías detectadas pueden hacer una investigación de la anomalía y poder descifrar cual fue la razón de la anomalía y así poder luego documentarlo y tener ese registro guardado.

10. Medidas de desempeño

1. **Anomalías detectadas:**
   Cantidad de anomalías detectadas al implementar el modelo, la idea es tener claro la cantidad de anomalías verdaderas se detectaron al igual que comparándolo con la cantidad de anomalías que fueron falsos positivos, y así poder compararlos y ver la efectividad del modelo.

2. **Tiempo utilizado en el preprocesamiento de datos:**
   La idea de detectar las anomalías tempranamente es poder tener una data mas limpia antes de comenzar a trabajar en ella, por lo que se quiere observar cuanto menos trabajo se debe realizar a la hora de hacer el preprocesamiento de datos (limpieza de datos).

3. **Precisión de los modelos de forecasting:**
   Comparar la precisión de los modelos de forecasting anteriores y ver si los modelos que se realizaran mas adelante con la data mas limpia mejoran su precisión en las distintas métricas que existen para medir esto, como el MAPE, MSE, MAE, etc.

4. **Tiempo de respuesta del sistema de detección de anomalías:**
   En un mundo ideal el modelo funcionaria sin problemas y no se atrasaría nunca a la hora de detectar una anomalía, por eso se quiere evaluar cuanto tiempo tarda el modelo en detectar una anomalía y luego cuando demora en reportarse esa anomalía.

5. **Estabilidad y robustez de los modelos de forecasting:**
   Medidas como la estabilidad de los modelos a lo largo del tiempo, especialmente en entornos cambiantes, al igual que observar su capacidad de adaptarse a nuevos datos

sin requerir ajustes frecuentes en el modelo.

11. Conclusión

Con la solución propuesta se logro obtener la primera iteración de un modelo que es capaz de detectar de buena manera algunas anomalías, pero al no considerarse el caudal a la hora de entrenar el modelo, se pudo notar que algunas de las anomalías detectadas eran simplemente por cambios en la amplitud de los datos, cuando si lo comparamos con el comportamiento del caudal en la figura 30 que esta en el anexo, podemos ver que dependiendo del caudal es como se mueve el nivel por lo que muchas de las anomalías detectadas en realidad no eran anomalías si no que eran comportamientos acordes a el comportamiento del caudal. Es por eso que se le propuso a la empresa realizar otro modelo el cual incluya el caudal en el entrenamiento, y luego buscar otras variables que puedan ayudar al modelo a ser mas preciso. Considerando que el modelo aun no es implementado en la empresa, y que todo se a realizado de manera local, no se han podido notar impactos en los datos de la empresa. Aun que por otro lado si el modelo se hubiese implementado tomaría un tiempo largo para comenzar a notar cambios en la calidad de los datos.

12. Referencias

Referencias

[1] Saxena, A. (2023). Anomaly Detection On Time Series Data - LSTM Autoencoder. Recuperado de https://github.com/alind-saxena/Anomaly_Detection/blob/main/Data%20Science/Anomaly%20Detection%20On%20Time%20Series%20Data%20-%20LSTM%20Autoencoder.ipynb

[2] Deep-Learning-For-Hackers. (2019, 24 de noviembre). Time Series Anomaly Detection with LSTM Autoencoders using Keras in Python. Recuperado de https://curiousily.com/posts/anomaly-detection-in-time-series-with-lstms-using-keras-in-python/

[3] Crépey, Stéphane, et al. "Anomaly Detection in Financial Time Series by Principal Component Analysis and Neural Networks." *Algorithms* 15 (2022): 385. DOI: 10.3390/a15100385.

[4] Zhou, Y., Ren, H., Li, Z., Pedrycz, W. (2022). Anomaly detection based on a granular Markov model. *Expert Systems with Applications*, 187, 115744. ISSN 0957-4174. DOI: 10.1016/j.eswa.2021.115744. Disponible en: https://www.sciencedirect.com/science/article/pii/S0957417421011222

[5] Lavin, A., Ahmad, S. (2015). Evaluating Real-Time Anomaly Detection Algorithms – The Numenta Anomaly Benchmark. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*. Miami, FL, USA (pp. 38-44). DOI: 10.1109/ICMLA.2015.141.

[6] Melichov, M. (Sep 19, 2022). Time Series Anomaly Detection With LSTM AutoEncoder. Recuperado de https://medium.com/@maxme006/time-series-anomaly-detection-with-lstm-autoencoder-b13a4177e241

13. Anexos

13.1. Códigos

```python
#Train test split
train = df1.loc[df1['Fecha'] <= '2022-10-31 23:59:59']
test = df1.loc[df1['Fecha'] > '2022-10-31 23:59:59']
train.shape, test.shape
```

```python
#Data scaling
scaler = StandardScaler()
scaler = scaler.fit(np.array(train['Nivel']).reshape(-1, 1))

train['Nivel'] = scaler.transform(np.array(train['Nivel']).reshape(-1, 1))
test['Nivel'] = scaler.transform(np.array(test['Nivel']).reshape(-1, 1))
```

```python
#Create sequences
TIME_STEPS = 60
def create_sequences(X, y, time_steps=TIME_STEPS):
    """
    Create sequences of input features and target values for training an LSTM model.

    Parameters:
        X (DataFrame): Input features.
        y (Series): Target values.
        time_steps (int): Number of time steps to consider in each sequence (default: TIME_STEPS).

    Returns:
        Xs (ndarray): Array of input sequences.
        ys (ndarray): Array of corresponding target values.
    """
    Xs, ys = [], []
    for i in range(len(X) - time_steps):
        Xs.append(X.iloc[i:(i + time_steps)].values)
        ys.append(y.iloc[i + time_steps])

    return np.array(Xs), np.array(ys)

X_train, y_train = create_sequences(train[['Nivel']], train['Nivel'])
X_test, y_test = create_sequences(test[['Nivel']], test['Nivel'])
print(f'Training shape: {X_train.shape}')
print(f'Testing shape: {X_test.shape}')
```

Figura 10: Entrenamiento LSTM Autoencoder Code

13.2. Gráficos de anomalías

```python
from sklearn.model_selection import train_test_split
from sklearn.ensemble import IsolationForest

# Assume that df is your DataFrame and it includes columns 'Volumen', 'Nivel', and 'Caudal'
X = df[['Volumen', 'Nivel', 'Caudal']]

# Split the data into training and testing sets
X_train, X_test = train_test_split(X, test_size=0.2, random_state=42)

# Train the model on the training set
model = IsolationForest(n_estimators=50, contamination=0.1)
model.fit(X_train)

# Predict the anomalies in the test data
y_pred_test = model.predict(X_test)

# Convert the predicted values to a binary format (1 for anomalies, 0 for normal records)
y_pred_test = [1 if pred == -1 else 0 for pred in y_pred_test]

# Add the predictions to your test DataFrame
X_test['Anomaly'] = y_pred_test
```

```python
import plotly.graph_objects as go

# Assume that X_test is your test data and 'Anomaly' is the column with predicted labels
anomalies = X_test[X_test['Anomaly'] == 1]
normal = X_test[X_test['Anomaly'] == 0]
```

Figura 11: Isolation Forest Code

```python
from sklearn.neighbors import LocalOutlierFactor
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px


# create train-test split
train_data, test_data = train_test_split(df, test_size=0.2, shuffle=False)

# extract the columns you want to use for anomaly detection
X_train = train_data[['Volumen', 'Caudal', 'Nivel']]
X_test = test_data[['Volumen', 'Caudal', 'Nivel']]

# scale the data
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# fit the LOF model on the training data
lof = LocalOutlierFactor(n_neighbors=100, contamination=0.05)
y_pred_train = lof.fit_predict(X_train)

# create a scatter plot to visualize the anomalies
fig = px.scatter(train_data, x='Fecha', y=['Nivel'], color=y_pred_train)
fig.show()
```

Figura 12: Local Outlier Factor Code

```python
#Build LSTM Autoencoder model
model = Sequential()
model.add(LSTM(units=128, activation = 'relu', input_shape=(X_train.shape[1], X_train.shape[2])))
#model.add(Dropout(rate=0.2))
model.add(RepeatVector(X_train.shape[1]))
model.add(LSTM(units=128, activation = 'relu', return_sequences=True))
#model.add(Dropout(rate=0.2))
model.add(TimeDistributed(Dense(units=X_train.shape[2])))
model.compile(optimizer=keras.optimizers.Adam(learning_rate=0.001), loss='mse')
model.summary()
```

```python
#Train the model
history = model.fit(
    X_train, y_train,
    epochs=50,
    batch_size=32,
    validation_split=0.1,
    callbacks=[keras.callbacks.EarlyStopping(monitor='val_loss', patience=5, mode='min')],
    shuffle=False
)
```

```python
#Plot training and validation loss
plt.plot(history.history['loss'], label='Training loss')
plt.plot(history.history['val_loss'], label='Validation loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
```

```python
#Save and load the model

# model.save('../Models/lstm_autoencoder_MM.keras')
model = keras.models.load_model('../Models/lstm_autoencoder_MM.keras')
```

Figura 13: Modelo LSTM Code

```
X_train_pred = model.predict(X_train)
train_mae_loss = np.mean(np.abs(X_train_pred - X_train), axis=1)

plt.hist(train_mae_loss, bins=50)
plt.xlabel('Train MAE loss')
plt.ylabel('Number of samples')

#Flag negative values as anomalies with threshold
threshold_neg = 0

threshold = np.percentile(train_mae_loss, 95)  # Adjust the percentile value as per your requirement
for percentile in [90,91,92,93,94,95,96,97,98,99]:
    threshold = np.percentile(train_mae_loss, percentile)
    X_test_pred = model.predict(X_test)
    test_mae_loss = np.mean(np.abs(X_test_pred - X_test), axis=1)
    anomalies = test_mae_loss > threshold
    print(f'Percentile: {percentile}, Number of anomalies: {np.sum(anomalies)}')
    print(f'Threshold: {threshold}')


#print(f'Reconstruction error threshold: {threshold}')
```

```
#Predictions on test data
X_test_pred = model.predict(X_test, verbose=1)
test_mae_loss = np.mean(np.abs(X_test_pred - X_test), axis=1)

plt.hist(test_mae_loss, bins=50)
plt.xlabel('Test MAE loss')
plt.ylabel('Number of samples')
```

```
anomaly_df = pd.DataFrame(test[TIME_STEPS:])
anomaly_df['loss'] = test_mae_loss
anomaly_df['threshold'] = threshold
anomaly_df['anomaly'] = anomaly_df['loss'] > anomaly_df['threshold']
anomaly_df['threshold_neg'] = threshold_neg
```

Figura 14: LSTM Autoencoder Code

```python
from scipy import stats

# Perform STL decomposition
stl = STL(df['Nivel'], period=12*24)
res = stl.fit()

# Calculate z scores of residuals
z = np.abs(stats.zscore(res.resid))

# Identify outliers
outliers = np.where(z > 3)[0]

# Plot the decomposition and anomalies
fig, ax = plt.subplots(2,1, figsize=(15,10))
res.observed.plot(ax=ax[0], title='Observed')
res.trend.plot(ax=ax[1], title='Trend')
res.seasonal.plot(ax=ax[1], title='Seasonal')
res.resid.plot(ax=ax[1], title='Residual')

# Add anomalies to the plot
ax[0].scatter(df.index[outliers], df['Nivel'].iloc[outliers], color='red', label='Anomaly')
ax[0].legend()

plt.show()
```

Figura 15: STL Decomposition Code

```python
from sendgrid import SendGridAPIClient
from sendgrid.helpers.mail import Mail

def send_email(subject, message, from_email, to_email, sendgrid_api_key):
    message = Mail(
        from_email=from_email,
        to_emails=to_email,
        subject=subject,
        plain_text_content=message)
    try:
        sg = SendGridAPIClient(sendgrid_api_key)
        response = sg.send(message)
        print(response.status_code)
        print(response.body)
        print(response.headers)
    except Exception as e:
        print(str(e))
```

```python
for date, row in anomaly_df.iterrows():
    if row['anomaly']:
        well_name = os.path.splitext(os.path.basename(csv_file))[0]  # Get the well name from the CSV file name
        date_time = row['Fecha']  # Replace 'Fecha' with the actual column name of the date and time in your DataFrame
        send_email(
            subject="Anomaly Detected",
            message=f"An anomaly was detected in well {well_name} at {date_time}",
            from_email="tduring@alumnos.uai.cl",
            to_email="thomas.during@gmail.com",
            sendgrid_api_key=[redacted]
        )
```

Figura 16: Alerta Anomalías Code

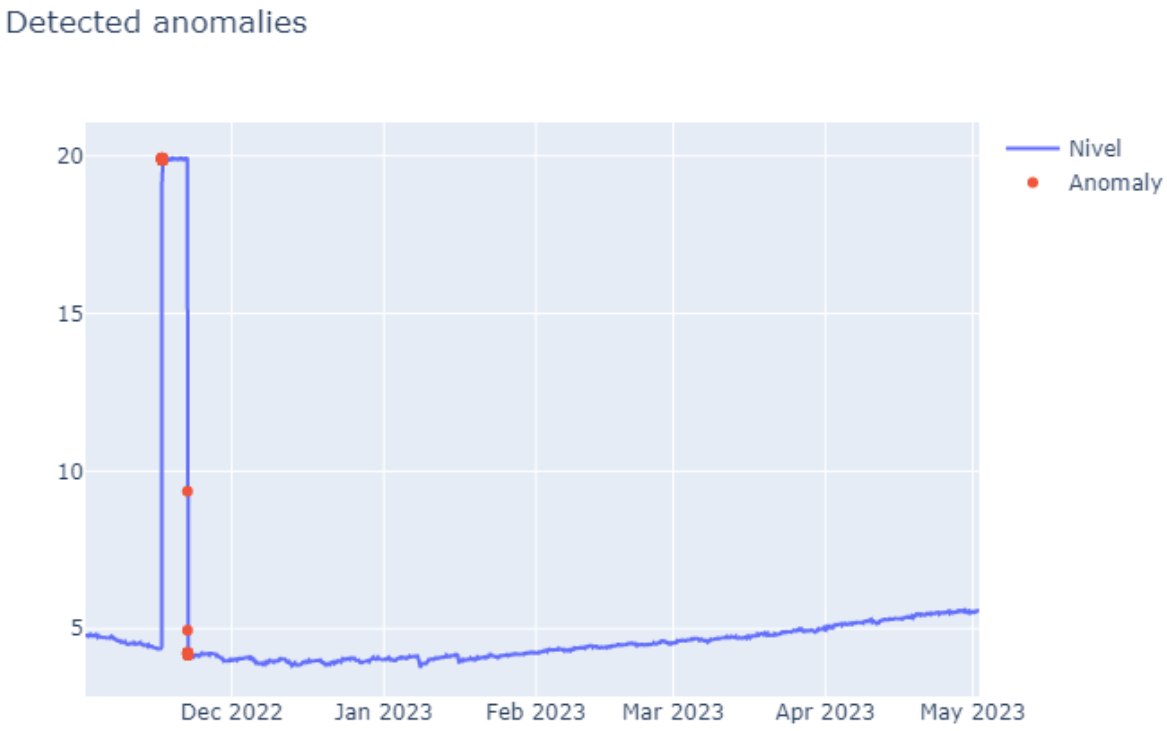

Figura 17: Gráfico Anomalías Pozo 1 El Melón

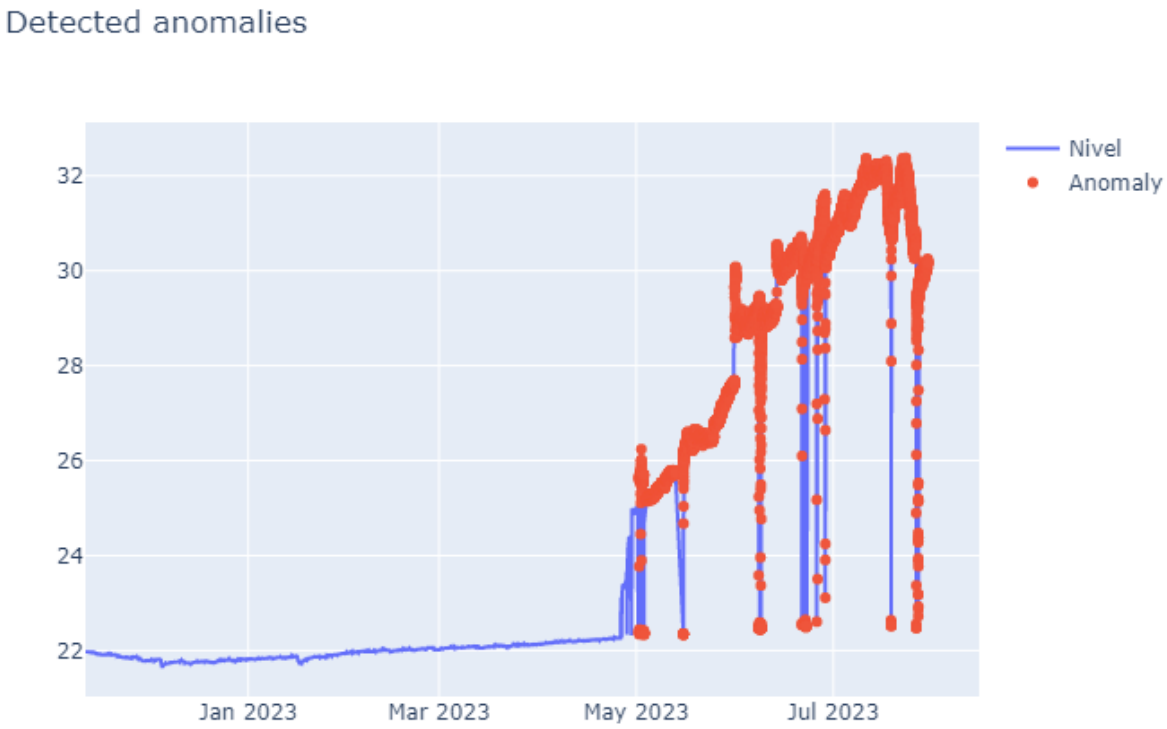

Figura 18: Gráfico Anomalías Pozo 2 El Melón

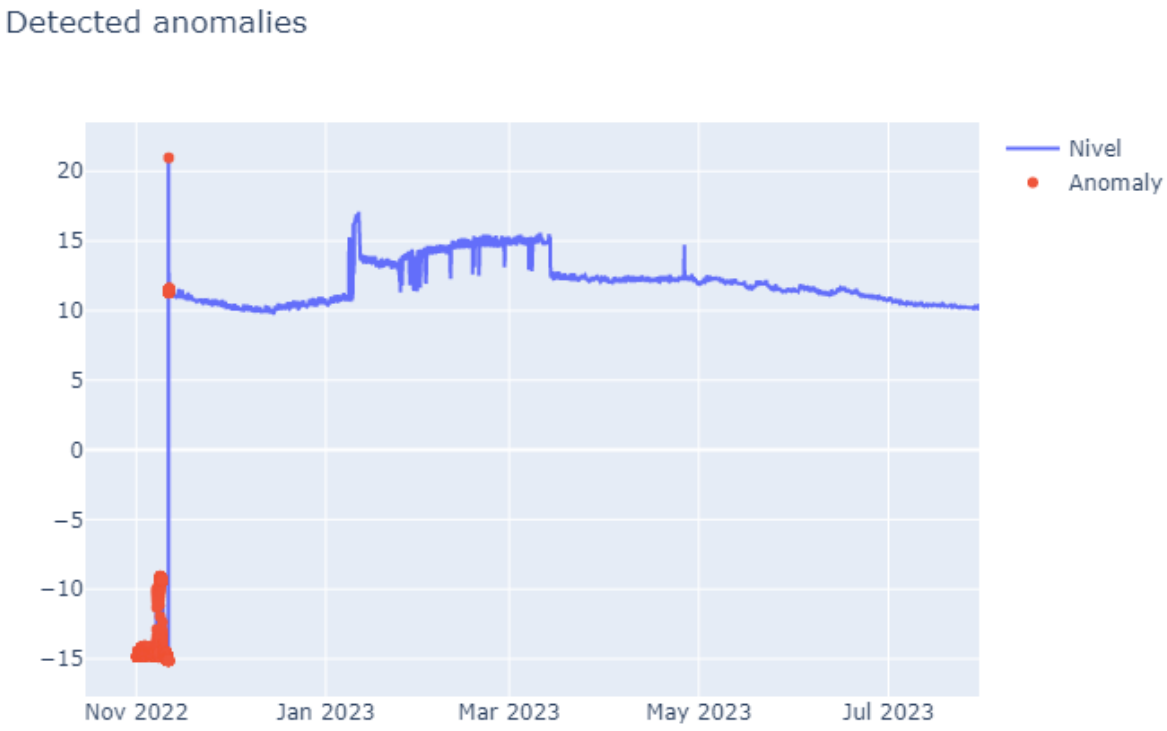

Figura 19: Gráfico Anomalías Pozo 3 El Melón

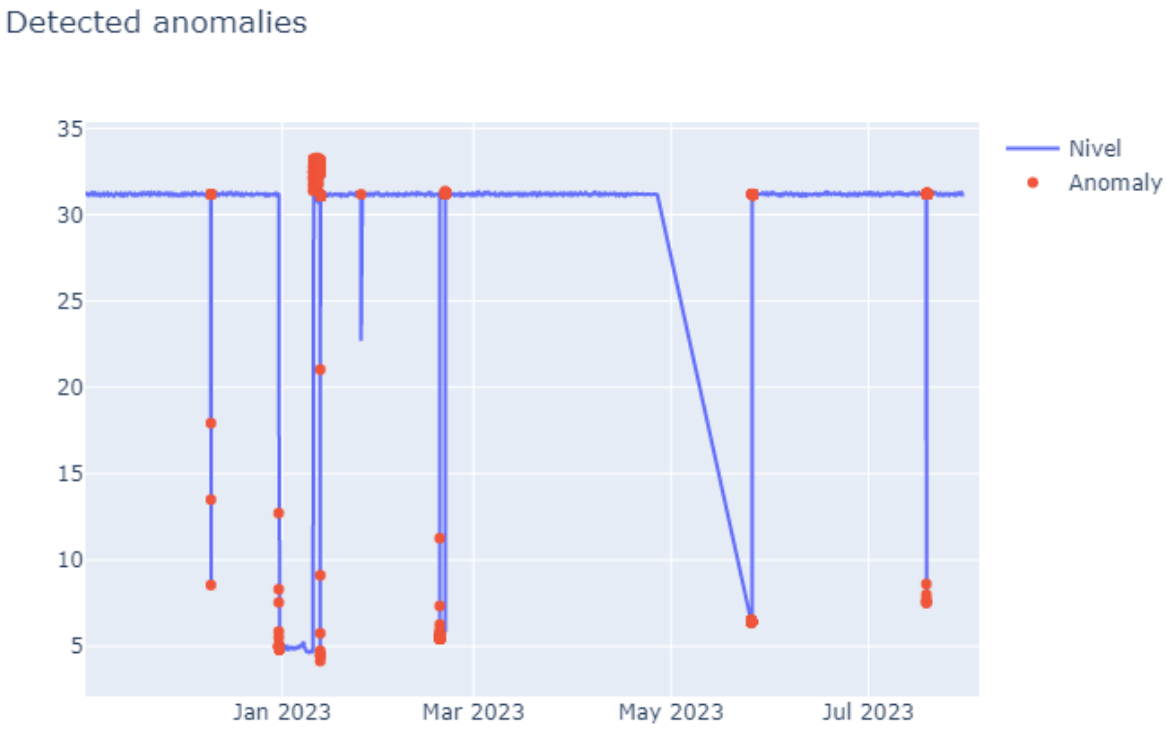

Figura 20: Gráfico Anomalías Pozo 4 El Melón

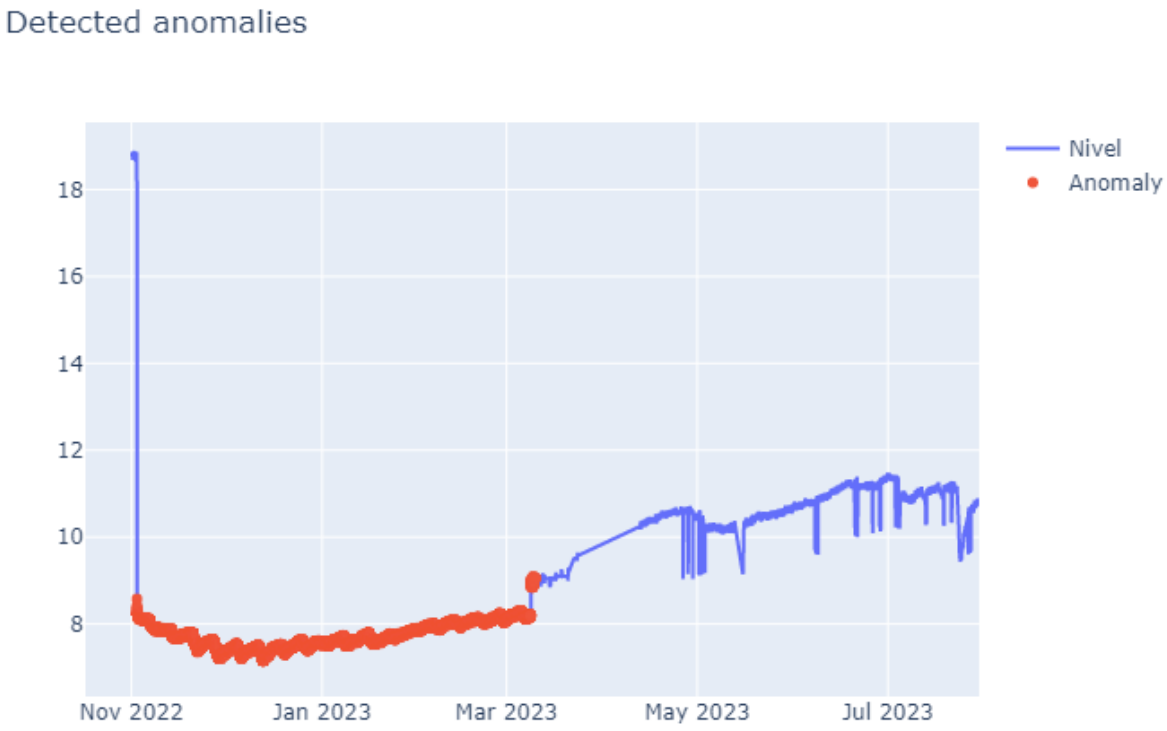

Figura 21: Gráfico Anomalías Pozo 6 El Melón

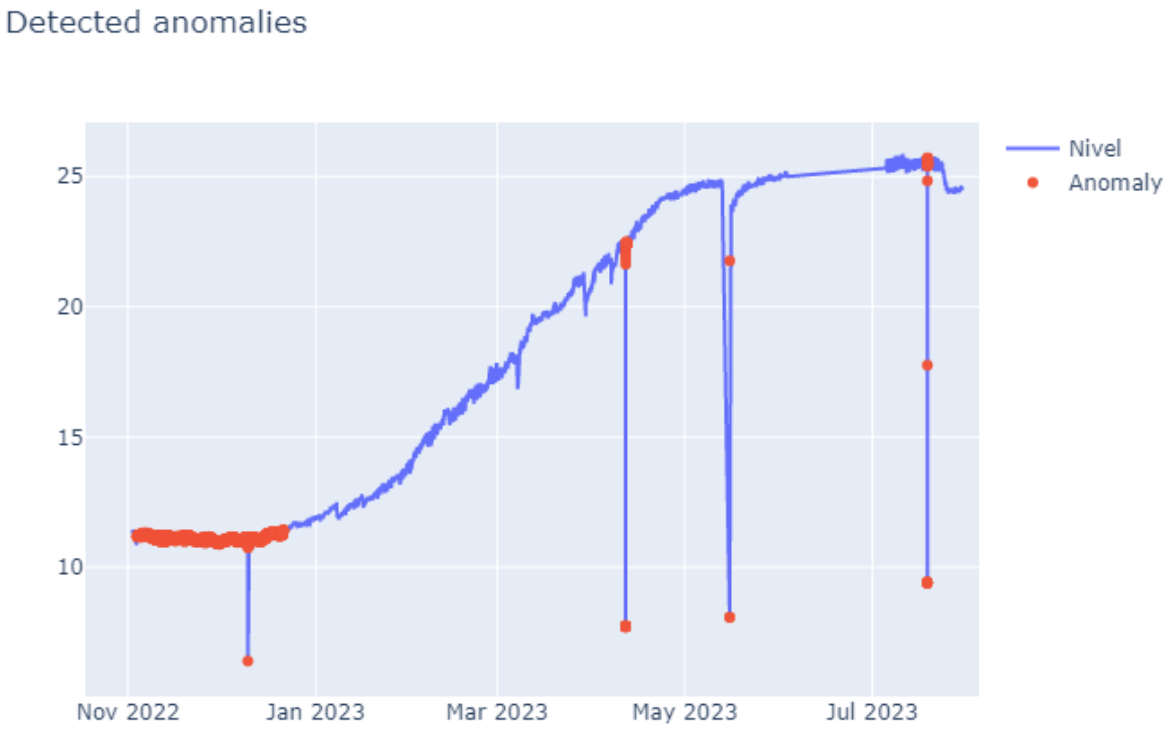

Figura 22: Gráfico Anomalías Pozo 9 El Melón

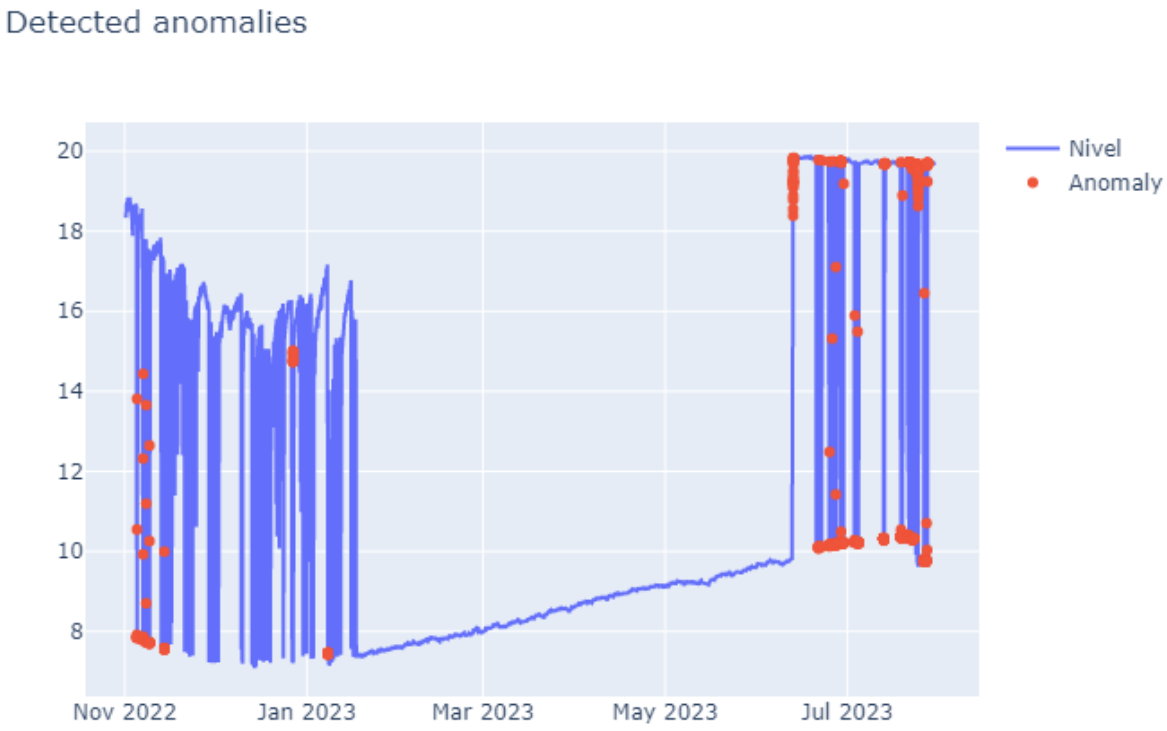

Figura 23: Gráfico Anomalías Pozo 10 El Melón

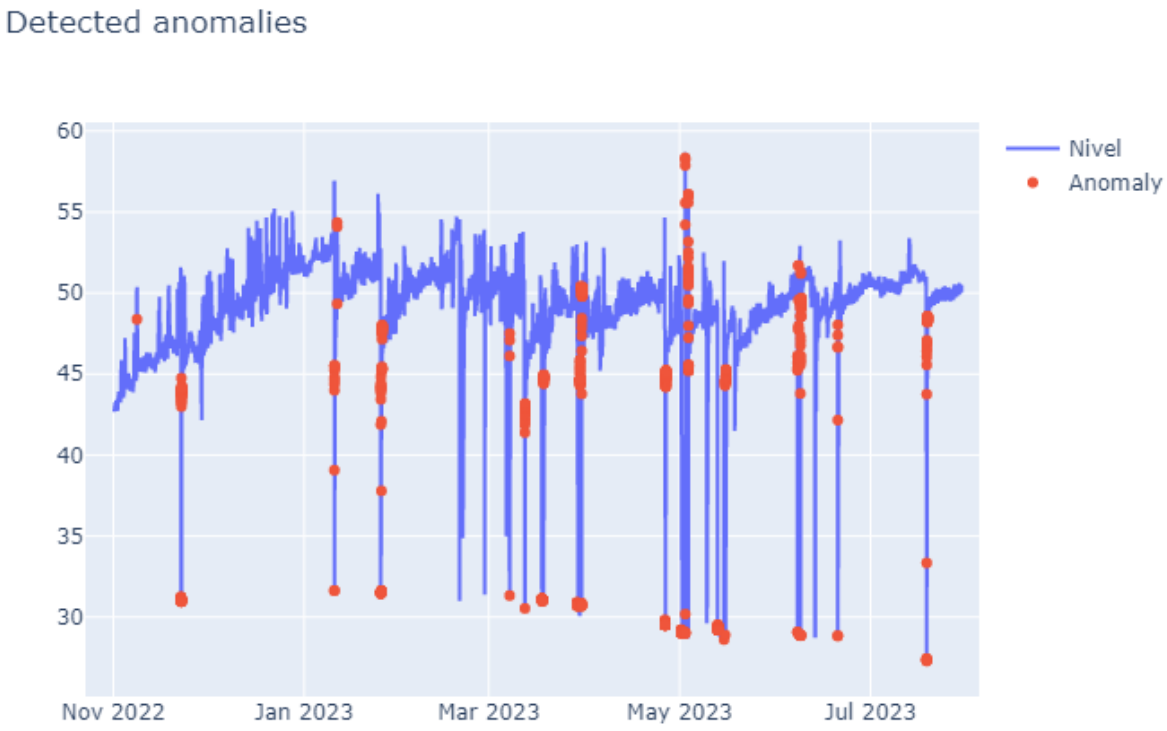

Figura 24: Gráfico Anomalías Pozo 1 Los Litres

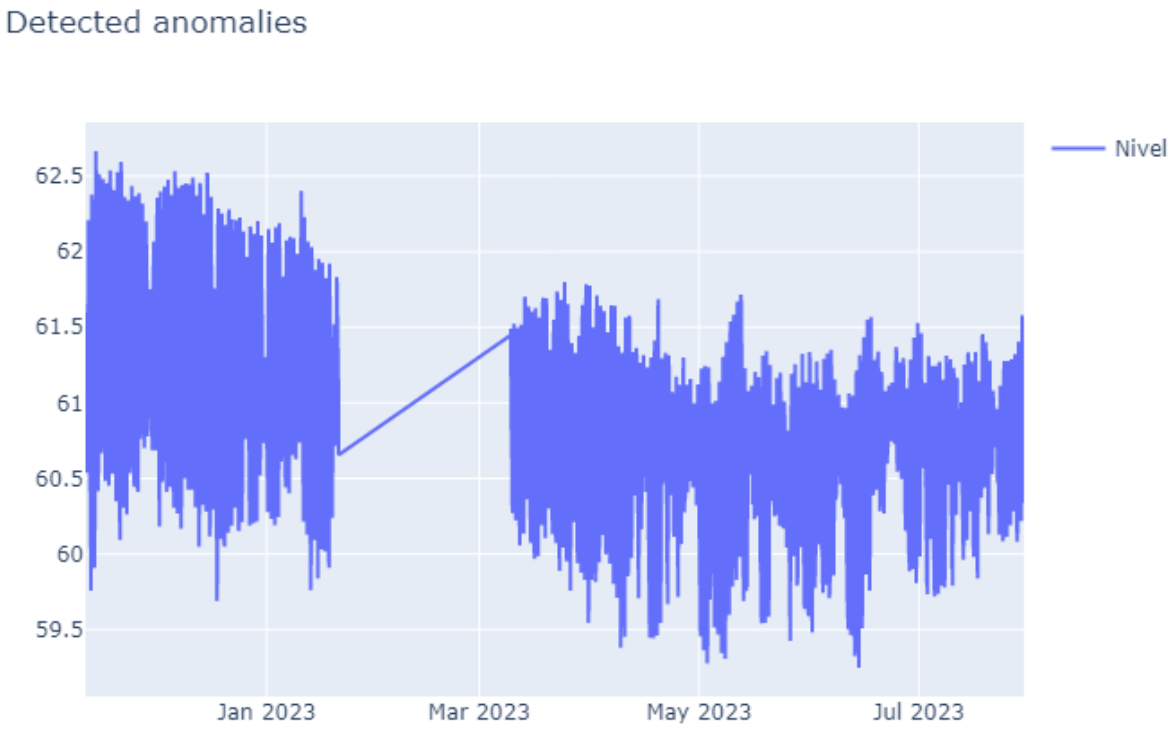

Figura 25: Gráfico Anomalías Pozo 2 Los Litres

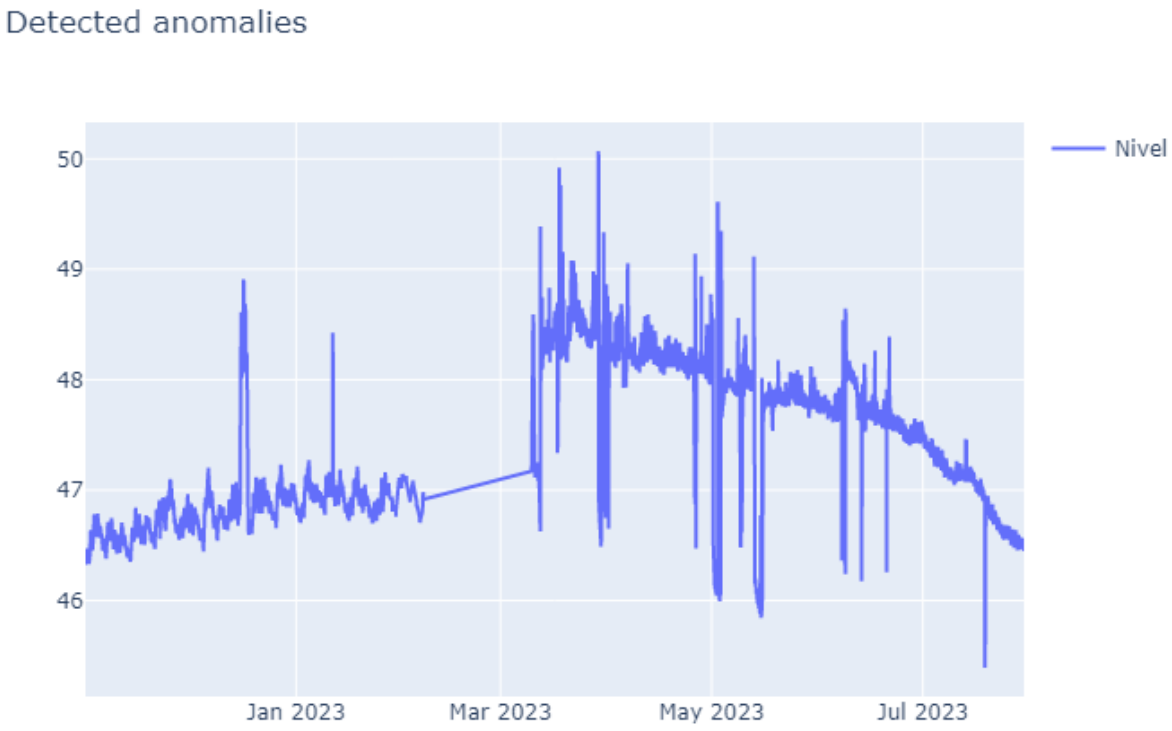

Figura 26: Gráfico Anomalías Pozo 3 Los Litres

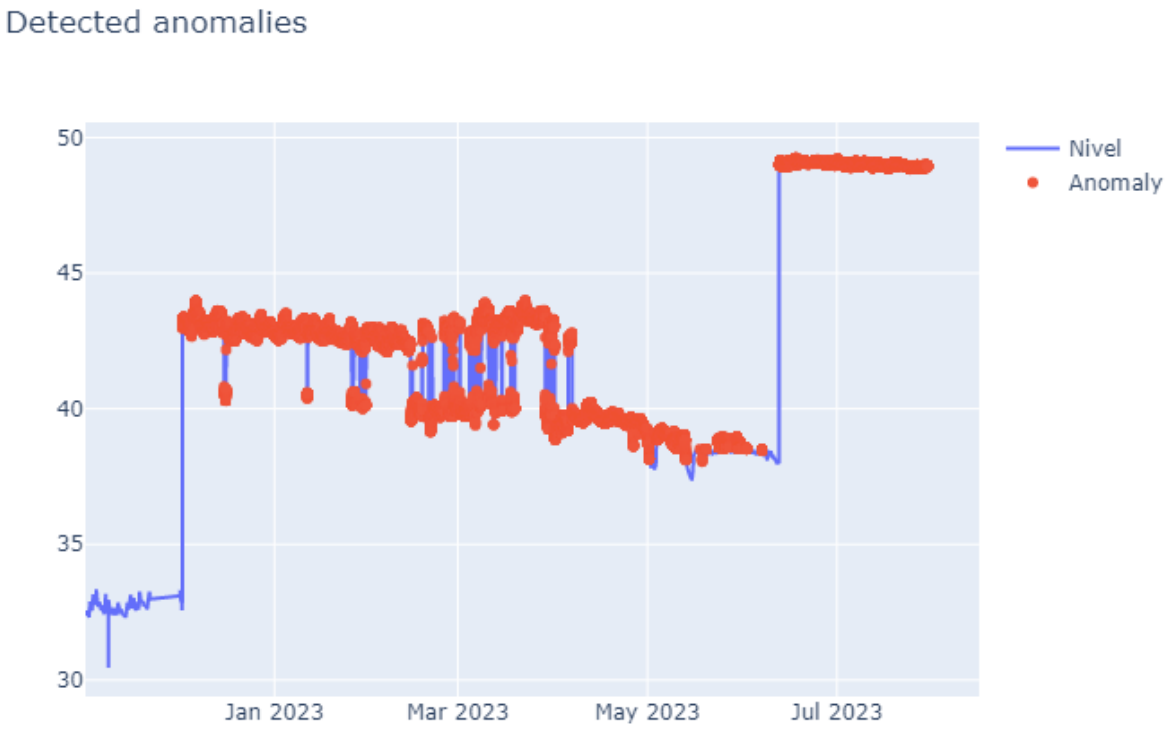

Figura 27: Gráfico Anomalías Pozo 4 Los Litres

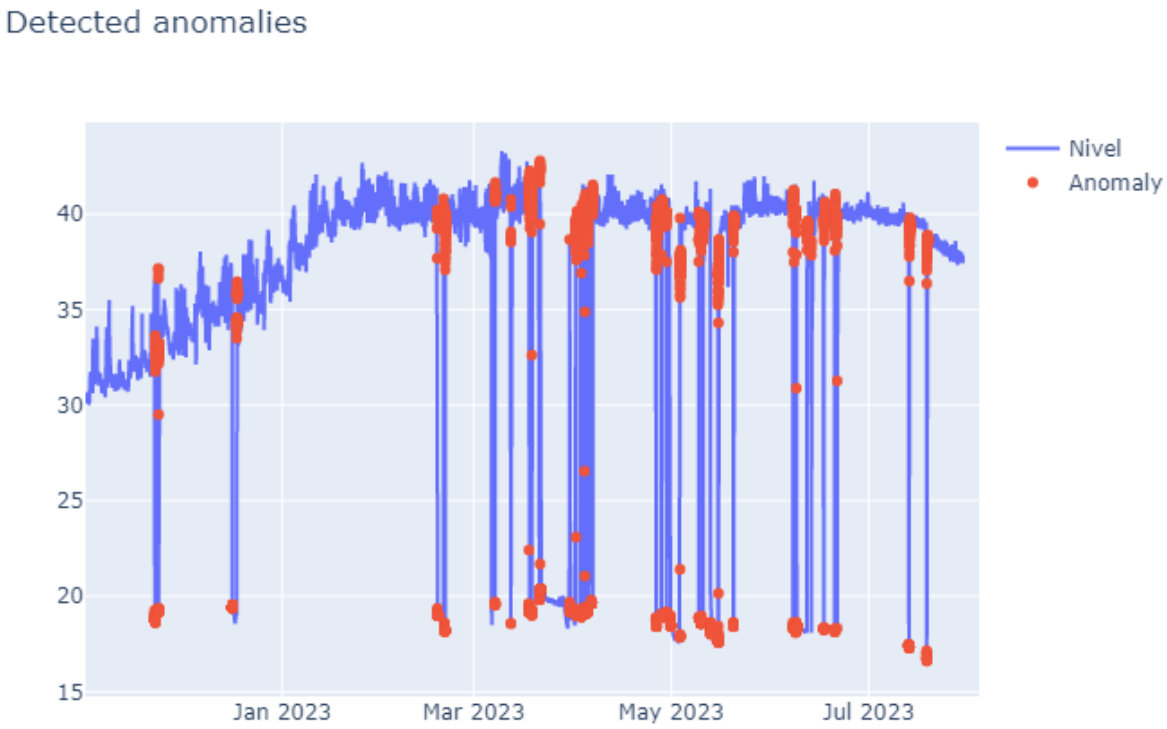

Figura 28: Gráfico Anomalías Pozo 5 Los Litres

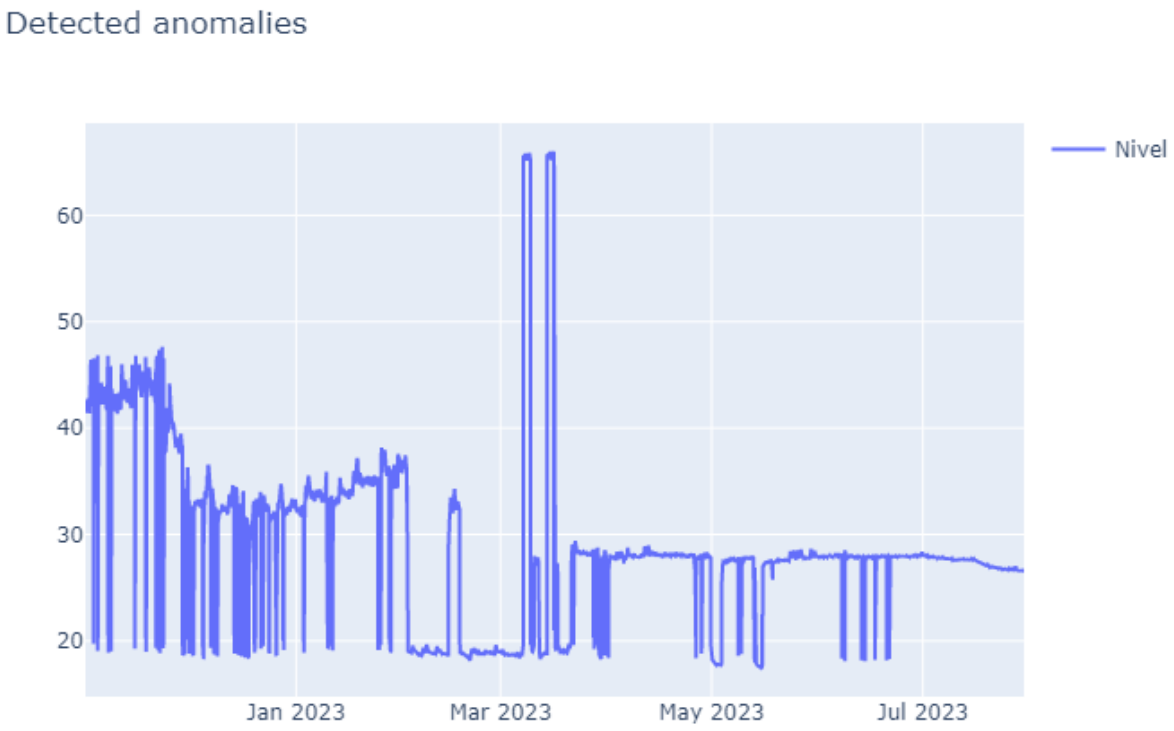

Figura 29: Gráfico Anomalías Pozo 6 Los Litres

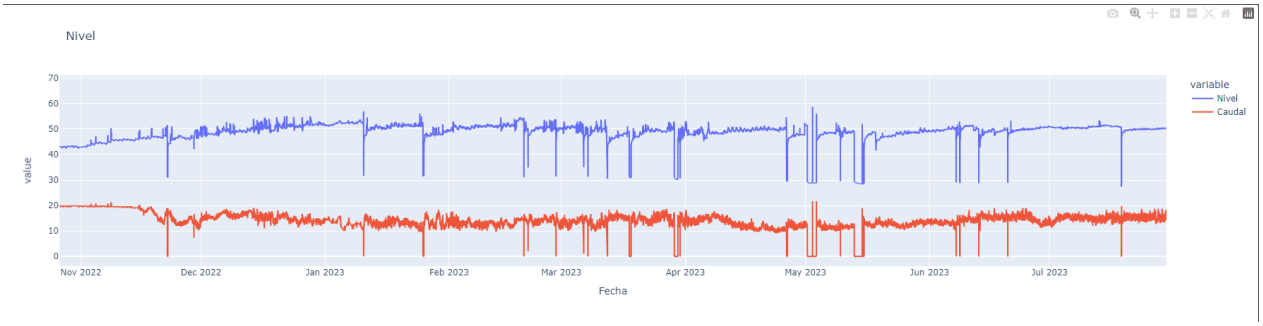

Figura 30: Comparación Nivel y Caudal en el tiempo