UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA MECÁNICA

DETECCIÓN DE ANOMALÍAS EN COMPONENTES MECÁNICOS EN BASE A DEEP LEARNING Y RANDOM CUT FORESTS

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL MECÁNICO

DIEGO ANDRÉS AICHELE FIGUEROA

PROFESOR GUÍA:
ENRIQUE ANDRÉS LÓPEZ DROGUETT

MIEMBROS DE LA COMISIÓN:
VIVIANA MERUANE NARANJO
JUAN TAPIA FARÍAS

SANTIAGO DE CHILE
2019

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE INGENIERO CIVIL MECÁNICO
POR: DIEGO ANDRÉS AICHELE FIGUEROA
FECHA: 2019
PROF. GUÍA: ENRIQUE ANDRÉS LÓPEZ DROGUETT

DETECCIÓN DE ANOMALÍAS EN COMPONENTES MECÁNICOS EN BASE A DEEP LEARNING Y RANDOM CUT FORESTS

Dentro del área de mantenimiento, el monitorear un equipo puede ser de gran utilidad ya que permite advertir cualquier anomalía en el funcionamiento interno de éste, y así, se puede corregir cualquier desperfecto antes de que se produzca una falla de mayor gravedad.

En *data mining*, detección de anomalías es el ejercicio de identificar elementos anómalos, es decir, aquellos elementos que difieren a lo común dentro de un set de datos. Detección de anomalías tiene aplicación en diferentes dominios, por ejemplo, hoy en día se utiliza en bancos para detectar compras fraudulentas y posibles estafas a través de un patrón de comportamiento del usuario, por ese motivo se necesitan abarcar grandes cantidades de datos por lo que su desarrollo en aprendizajes de máquinas probabilísticas es imprescindible. Cabe destacar que se ha desarrollado una variedad de algoritmos para encontrar anomalías, una de las más famosas es el *Isolated Forest* dentro de los árboles de decisión. Del algoritmo de *Isolated Forest* han derivado distintos trabajos que proponen mejoras para éste, como es el *Robust Random Cut Forest* el cual, por un lado permite mejorar la precisión para buscar anomalías y, también, entrega la ventaja de poder realizar un estudio dinámico de datos y buscar anomalías en tiempo real. Por otro lado, presenta la desventaja de que entre más atributos contengan los sets de datos más tiempo de cómputo tendrá para detectar una anomalía. Por ende, se utilizará un método de reducción de atributos, también conocido como reducción de dimensión, por último se estudiará como afectan tanto en efectividad y eficiencia al algoritmo sin reducir la dimensión de los datos.

En esta memoria se analiza el algoritmo *Robust Random Cut Forest* para finalmente entregar una posible mejora a éste. Para poner en prueba el algoritmo se realiza un experimento de barras de acero, donde se obtienen como resultado sus vibraciones al ser excitado por un ruido blanco. Estos datos se procesan en tres escenarios distintos: Sin reducción de dimensiones, análisis de componentes principales(*principal component analysis*) y autoencoder. En base a esto, el primer escenario (sin reducción de dimensiones) servirá para establecer un punto de orientación, para ver como varían el escenario dos y tres en la detección de anomalía, en efectividad y eficiencia.

En los resultados se observa una mejora al reducir las dimensiones en cuanto a tiempo de cómputo (eficiencia) y en precisión (efectividad) para encontrar una anomalía, finalmente los mejores resultados son con análisis de componentes principales (*principal component analysis*).

Para Felipe Andres Zersi Márquez (Q.E.P.D).

Agradecimientos

Hola. Finalmente terminé una etapa de mi vida que empezó en la enseñanza media, la cual no fue un tiempo de soledad por lo que quiero agradecer:
A mi familia, padres y hermanos por aconsejarme, cuidarme y apoyarme. Y a mis abuelos por regalonearme.
A mi polola, Ivana M. con sus palabras y cariño, que me ha apoyado siempre, por estar siempre ahí y hacerme ver que soy capaz de muchas cosas.
A mis amigos del colegio, Cata, Maca, Cristian y Gonzalo, que fue con los que más compartí en la enseñanza media.
A los amigos que conocí en la EDV y seguí compartiendo con ellos durante la universidad, Javi Benito, Felipe A. y José Luis M. (Pollo).
También al grupo de amigos Grupote.
A los amigos de El Salvador (el casco romano) que siempre me recibían en sus juntas, me alegra ver que su amistad se ha ido fortaleciendo desde que iban en el colegio Víctor, Karen y Pochi.
Agradezco a todos los integrantes de La Radio Integral (Eduardo, Andrés, Pedro, Claudio, Eva, Bastián, Mauricio, Sebastián, Diego G, Jimmy, Panchito, Nicolás, Quezada, Cobo...) compartir con ustedes en la U, en las transmisiones en vivo, las grabadas y en la salita 8, de aquí nacieron muchas amistades, en verdad se aprende mucho de este grupo organizado.
A las amistades de Sección 8, Mahu, Maxi Padilla y Fran Oporto y Levet (aunque no eres de la 8).
A los amigos de mecánica, Fernanda, Carlos Poblete, Yanara, Guille, Vicky O, Felipe T, Leo-San, Nachín y Cristian O.
Al Drim Tim de trabajos en mecánica Nicolás M. y Cata B. que de alguna forma siempre salíamos trabajando y sin ningún problema terminábamos nuestros trabajos.
Agradezco a los amigos industriales que se pelean a cada rato Alan y Carlos.
A mis amigos que estudian/estudiaron afuera, pero no por eso nos dejamos de hablar Rubén R, FAP y Cristian G. (Pedantic Anarchy). De verdad, agradezco mucho por haber tenido ramos con ustedes y haber seguido compartiendo con ustedes, aunque los ramos hayan terminado las amistades no terminaban. Gracias a Mohammad por orientarme en el desarrollo de la memoria. Finalmente agradezco a la persona que está leyendo esto que de alguna forma u otra, leyó hasta el final de mis agradecimientos por estar buscándose y olvidé mencionar (culpa de dejar los agradecimientos al final de la entrega(??)) y por otra parte, gracias al que simplemente descargó la cybertesis, por estar interesado en este tema.

Tabla de Contenido

Índice de Tablas

Índice de Ilustraciones

Introducción

En la actualidad existe mayor acceso a grandes volúmenes de datos, los que se pueden obtener a través de sensores de vibración, sensores de temperatura e incluso sensores ópticos como el de una cámara. Esto constituye una gran oportunidad para mejorar las técnicas de mantenimiento de una estructura o equipo.

Actualmente se ha desarrollado una variedad de algoritmos para encontrar anomalías, una de las más famosas es el *IF*. Del algoritmo de IF han derivado distintos trabajos que proponen mejoras para éste cómo es el *Robust Random Cut Forest* el cual, además de mejorar su presición para buscar anomalías[1], también entrega la ventaja de poder realizar un estudio dinámico de datos y buscar anomalías en tiempo real. Esta propiedad de RRCF será de mucha utilidad para un monitoreo de equipos a lo largo del tiempo por lo que se selecciona RRCF para la memoria. Según la bibliografía observada es importante destacar que existe la desventaja que entre más atributos contengan los sets de datos, más tiempo de cómputo tendrá para detectar una anomalía[2]. Se sugiere utilizar un método de reducción de atributos, también conocido como reducción de dimensión.

Mediante los datos de vibración de una estructura se puede alimentar un modelo de redes neuronales y luego utilizar el algoritmo *Robust Random Cut Forest* (RRCF) [1] para la detección de anomalías semi-supervisado en componentes mecánicos.

Para poder analizar la efectividad de RRCF para la detección de anomalías semi-supervisado, se realizará un ensayo experimental que se desarrollará en el "Laboratorio de vibraciones" del Departamento de Ingieniría Civil Mecánica de la Universidad de Chile. El experimento consiste en obtener datos de vibración de una barra de acero sana de 85x2,5x1,5 cm y tres barras dañadas de las mismas dimensiones. El daño de cada barra corresponde a un agujero de distintas profundidades, ubicadas a 25 cm de uno de los extremos. Tras la obtención de los datos de vibraciones, éstos se pasarán por un análisis de frecuencia, que corresponde a una técnica que permite obtener datos representativos en frecuencia sin la necesidad de una caracterización completa en el dominio temporal, utilizando por ejemplo la transformada de Fourier[3]. Los datos atemporales obtenidos pasarán por una reducción de parámetros de redes neuronales denominado *Autoencoder*[4] y *Principal Component Analysis* (PCA) [5], para finalmente utilizar RRCF y analizar la efectividad de éste para distintos métodos de reducción de dimensionalidad.

Capítulo 1

Motivación

El área mantenimiento ha sido un tema de gran interés para el alumno, por esta razón la presente memoria se orienta en la búsqueda de una forma efectiva y eficiente de detectar anomalías de manera semi-supervisada, es decir, se tiene la intención de entrenar un modelo matemático que sea capaz de admitir nuevos datos e identificar si estos son anómalos o no con el fin de revisar un equipo mecánico periódicamente, hasta poder detectar cuando su funcionamiento se encuentra fuera de su rango del normal. Por consiguiente, la motivación del alumno es utilizar el algoritmo RRCF para detección de anomalías con datos de un ensayo experimental de vibraciones de barras. Primero se verá la efectividad de RRCF para detectar anomalías, luego se espera que al utilizar un método de reducción de dimensión esta efectividad aumente.

1.1. Objetivos

1.1.1. Objetivo general

Analizar la efectividad de RRCF en conjunto con métodos de reducción de dimensión para la detección de anomalías semi-supervisado.

1.1.2. Objetivos específicos

- Generar datos de vibración mediante un ensayo de experimental de barras.
- Poner a prueba RRCF con distintos sets de datos y procesar los datos de ensayos experimentales para adaptar al algoritmo mencionado.
- Optimizar resultados con cada método de reducción de dimensión propuesto.

1.2. Alcance

Se estudiará la efectividad de RRCF para la detección de anomalías semi-supervisado, con métodos de reducción de dimensión, donde se espera mejorar su rendimiento para detección de anomalías.

1.3. Organización de texto.

En esta sección se explicará la organización del texto para poder orientar al lector.

En el capítulo II se realizará la revisión bibliográfica sobre Robust Random Cut Forest y además se explicará cómo utilizarlo.

En el capítulo III se describirán aspectos vinculados al experimento realizado, tanto de los equipos como del software utilizado para los sensores y el montaje del experimento.

En el capítulo IV se explicará la metodología de trabajo utilizada y los mecanismos de análisis de datos.

En el capítulo V se mencionarán los resultados obtenidos.

Finalmente, en el capítulo VI se presentarán las conclusiones del trabajo realizado.

Capítulo 2

Revisión Bibliográfica

Al observar la Figura 2.1 y suponiendo que se debe responder la siguiente pregunta: ¿cuántas almendras hay en el envase? La respuesta debe ser entregada basándose únicamente en el poder de la observación, sin la capacidad de abrir el envase y contar las almendras una por una.

Figura 2.1: Envase con almendras. Fuente: Elaboración propia.

Si la pregunta anterior se le realizara a mil personas y luego sus promediaran sus respuestas en varios casos esta respuesta promedio será más exacta que la respuesta de algún experto (en almendras, en el ejemplo). A este fenómeno se le llama 'Inteligencia Colectiva'[6]. En

aprendizaje de máquinas probabilístico existen algoritmos como los Bosques Aleatorios (RF, Random Forest) que se basan en este fenómeno. De la manera en que ensamblan n árboles que puedan dar una predicción ya sea de clasificación o regresión para un problema, pero la respuesta final del algoritmo es el promedio de los n árboles que se ensamblaron. RF es uno de algoritmos de aprendizaje más certeros que hay disponible. Lo que genera que existan otros algoritmos de aprendizaje basados en la 'Inteligencia Colectiva' pero con la capacidad de detectar anomalías como *Isolated Forest*(IF)[7].

Observación Si se tiene una base de datos de hojas con los atributos de: Largo de hoja, ancho de hoja, color de hoja y planta de la que proviene, a cada uno de estos atributos se les llamara Dimensión. Esto quiere decir que la base de datos de hojas contiene una dimensión igual a 4.

Esta memoria se centra en la detección de anomalías semi-supervisadas, buscando bibliografía se encontró el algoritmo RRCF que es una mejora del algoritmo IF[1]. El cambio más importante es que en IF al generar un árbol aleatorio éste selecciona una dimensión de manera uniformemente aleatoria, mientras que en RRCF la selección de la dimensión es proporcional al tamaño de cada dimensión, por consecuencia en RRCF las 'dimensiones más chicas', también denominadas dimensiones irrelevantes', son ignoradas llegando a tener mejores resultados en detección de anomalías.

No obstante, una de las desventajas conocidas de RRCF es que al aumentar las dimensiones de una base de datos, el tiempo de cómputo aumenta proporcionalmente como se puede ver en la Figura 2.2.

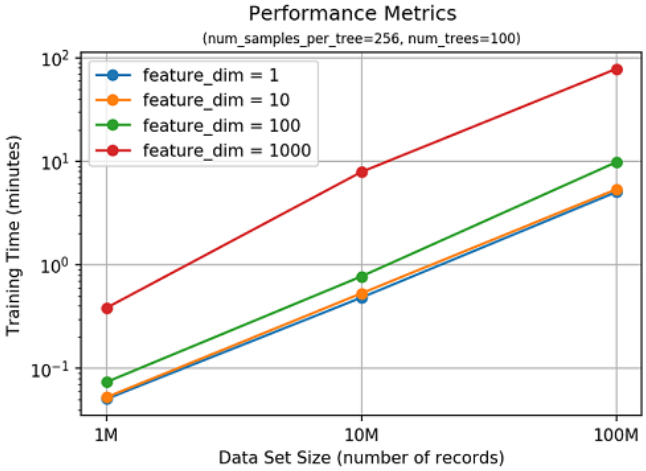


Figura 2.2: Rendimiento de tiempo de cómputo en función del número de dimensiones. Fuente: [2].

Además, como se señaló en la motivación, el estudiante se centra en detección de anomalías semi-supervisado, aunque una de las diferencias se puede dar por los etiquetados tal como se muestra en la Figura 2.3. El alumno al seleccionar la detección de anomalías semi-supervisada se condiciona a entrenar un modelo solamente con los datos sanos, es decir, sin anomalías.

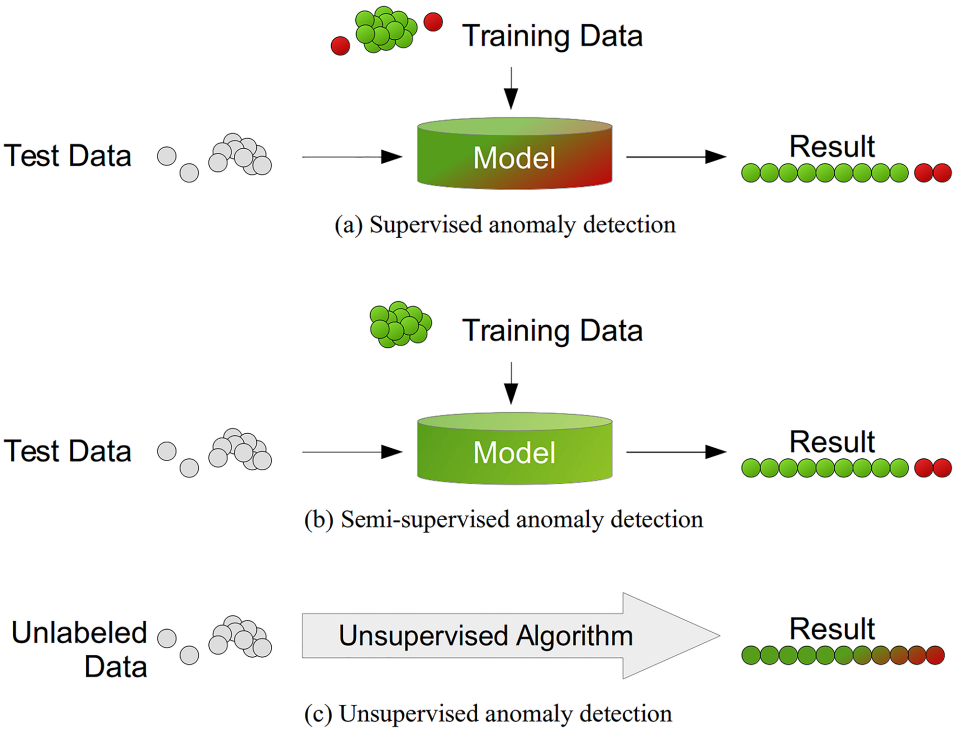


Figura 2.3: Diferentes modos de detección de anomalías dependiendo de la disponibilidad de los etiquetados en datos. Fuente: [8].

Ahora en la Sección 2.1. se hablará del paper de RRCF, cuya explicación es relevante para poder entender mejor la forma de tratar los datos.

2.1. Robust Random Cut Forest Based Anomaly Detection On Streams

La detección de anomalías es una piedra angular en problemas de *data mining*. A pesar de que el problema de detección de anomalías ha sido bien estudiado en las últimas décadas, la emergente expansión de los datos de internet de las cosas (Internet of Thing, IOT) y los sensores conlleva a considerar el problema desde una perspectiva de señales. Para poder estudiar esta perspectiva se debe responder principalmente a dos preguntas:

1. ¿Cómo se define una anomalía?
2. ¿Qué estructura se puede usar para detectar anomalías en datos de señales dinámicos?

Para la primera pregunta, si se ve desde el punto de la complejidad de un modelo o de una muestra de datos, se puede definir un punto anómalo si al insertar este punto al modelo se incrementa substancialmente la complejidad tratar los datos.

Mientras que, para la segunda pregunta, se propone un bosquejo de un modelo denominado formalmente RRCF. Se define de la siguiente manera:

Definición 2.1 (ver [1]) *Un árbol de corte aleatorio robusto, Robust Random Cut Tree (RRCT) de un conjunto de datos S es generado por los siguientes pasos:*

1. *Se elige una dimensión de manera aleatoria proporcional a* $\frac{l_i}{\sum_j l_j}$*, donde* l_i *es la distancia de los dos puntos más lejanos de la dimensión* i*, es decir,* $l_i = max_{x \in S} x_i - min_{x \in S} x_i$.

2. *Se elige un* $X_i \sim Uniforme[min_{x \in S} x_i, max_{x \in S} x_i]$.

3. *Define* $S_1 = \{x | x \in S, x_i \leq X_i\}$ *y* $S_2 = S - S_1$ *y luego aplicar esta definición recursivamente en* S_1 *y* S_2

Finalmente un RRCF es el conjunto de RRCT.

De la definición anterior se puede concluir el primer teorema.

Teorema 2.2 (ver [1]) *Si se considera el peso de un nodo en un árbol como la suma de dimensiones* $\sum_i l_i$*. Y se definen dos puntos u y* $v \in S$*, la distancia en un árbol de un punto u a v se define como el peso del nodo 'padre' que une a ambos. Entonces la distancia es al menos la distancia de Manhattan* $L_1(u, v)$ *y a lo más* $O\left(d \log \frac{|S|}{L_1(u,v)}\right)$ *veces* $L_1(u, v)$.

Lo interesante de este teorema es que si un punto x se encuentra lejos de otros puntos (como en el caso de un punto anómalo), se espera que, en un RRCT, este punto x se encuentre más lejos, ya que la distancia de Manhattan es mayor o igual a una distancia euclidiana como se puede ver en el apéndice 6.2.

La selección beneficia aquellas dimensiones que tienen mayor variación en los valores de su dimensión por lo que al tener muchas dimensiones no relevantes, es decir, dimensiones que tienen baja variación de sus valores, la comparación del algoritmo genera más tiempo de búsqueda y en consecuencia, el algoritmo se beneficia al trabajar a bajas dimensiones.

En este algoritmo se tiene el interés de mantener la producción de árboles aleatorios con datos de manera dinámica, por lo que se propone una forma de extraer un dato del árbol ($\mathcal{T}(S - x)$) y a la vez de la inserción de un dato ($\mathcal{T}(S \cup x)$).

Teorema 2.3 (ver [1]) *Dado un árbol T dibujado con los de una muestra S se tiene* $\mathcal{T}(S)$*; si se borra un nodo que contiene el punto anómalo x y a su padre, el hermano del punto anómalo se puede ajustar a su nodo 'abuelo' (como se muestra en la Figura 2.4), entonces el árbol resultante es* T' *tiene la misma probabilidad de ser dibujado como* $\mathcal{T}(S - x)$.

En el teorema 2.3 se puede ver naturalmente con la Figura 2.4 donde la extracción de x conlleva que el árbol disminuya un piso. En efecto, si se inserta x se necesitará realizar unas operaciones más que al extraerlo.

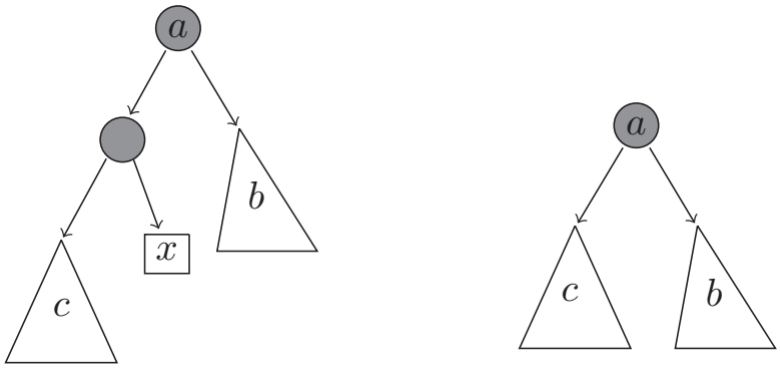


Figura 2.4: Mantenimiento de reducción de los árboles. Izquierda árbol T y derecha árbol T'. Fuente: [1].

Teorema 2.4 (ver [1]) *Se puede mantener un RRCT sobre una muestra S incluso si la muestra se actualiza dinámicamente para datos transmitidos por streaming usando un tiempo de actualización sublineal y espacio $O(\mathrm{d}|S|)$.*

Teorema 2.5 (ver [1]) *Se puede mantener un RRCT sobre una muestra S aun cuando la muestra S se encuentra actualizándose dinámicamente usando una actualización del tiempo y espacio de $O(\mathrm{d}|S|)$*

Teorema 2.6 (ver [1]) *Dado un RRCT $T(S)$ para datos S, si el árbol pierde una altura producto de la eliminación de un dato, entonces se debe considerar un algoritmo que reconstruya un árbol equivalente con una altura menos.*

Teorema 2.7 (ver [1]) *Dado un RRCT $T(S)$ para datos S y punto p, es fácil calcular un RRCT $T(S+p)$ siguiendo el algoritmo descrito en la definición 1.*

2.1.1. Definiendo una anomalía

La tarea fundamental es cuantificar el cambio que produce la presencia de un punto anómalo. Para eso se asignan a las ramas de la izquierda el bit 0 y a las ramas de la derecha el bit 1. Se puede observar la complejidad de un punto en un árbol, depende del recorrido de bit que se tiene para llegar a éste. Ahora, si se tiene un set de puntos Z y un punto $y \in Z$ entonces $f(y, Z, T)$ será la función que defina la profundidad del punto y en el árbol T. Considerando que se utiliza el teorema 2.3 en un árbol $T(Z - \{x\})$. Notar que dato T y x es únicamente determinado. Entonces la profundidad del punto y en $T(Z - \{x\})$ se determinará como $f(y, Z - \{x\}, T)$.

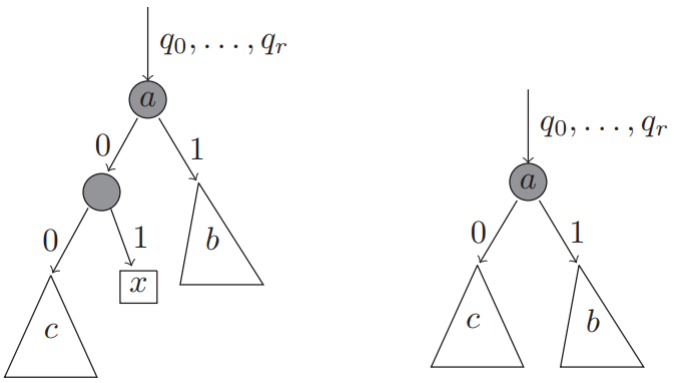


Figura 2.5: Izquierda árbol $T(Z)$ y derecha árbol $T(Z-x)$. Fuente: [1].

Ahora, si se considera que el punto y pertenece al subárbol c de la Figura 2.5. El bit que representa a y en T(de la izquierda) será $q_0, ..., q_r, 0, 0, ...$ y que luego en el árbol derecho sería $q_0, ..., q_r, 0, ...$ La complejidad se denotará como $|M(T)|$ el número de bits requeridos para describir todos los puntos y en el árbol T por lo tanto $|M(T)| = \sum_{y \in Z} f(y, Z, T)$. Si se remueve x entonces el nuevo modelo de complejidad sería $|M(T')| = \sum_{y \in Z-\{x\}} f(y, Z - \{x\}, T')$. Donde $T' = (Z - \{x\})$ es un árbol sobre la muestra $Z - x$. Sin embargo gracias al teorema 2.3 existen diversas formas de mapear desde $T(Z)$ a $T(Z - \{x\})$ y se puede obtener:

$$\begin{aligned} &\mathbb{E}_{T(Z)}[|M(T)|] - \mathbb{E}_{T(Z-\{x\})}[|M(T(Z-\{x\}))|] = \\ &\sum_{T} \sum_{y \in Z-\{x\}} \mathbb{P}r[T]\left(f(y, Z, T) - f(y, Z-\{x\}, T') + \right. \\ &\qquad \left. \sum_{T} \mathbb{P}r[T] f(x, Z, T)\right. \end{aligned} \tag{2.1}$$

Definición 2.8 *Se define como el desplazamiento de un punto x al incremento de al modelo complejidad que realiza a los otros puntos. Es decir, para un set de datos Z, para calcular el desplazamiento producido por la extracción de x se define como:*

$$Disp(x, Z) = \sum_{T, y \in Z-\{x\}} \mathbb{P}r[T]\left(f(y, Z, T) - f(y, Z-\{x\}, T')\right) \tag{2.2}$$

Notar que el cambio total en el modelo de complejidad es $Disp(x, Z) + g(x, Z)$, donde $g(x, Z) =_T \mathbb{P}r[T] f(x, Z, T)$ sería la profundidad esperada del punto x en un modelo aleatorio. En lugar de postular la anomalía correspondiente al largo de $g()$, hay que centrarse en el largo del $Disp()$. El nombre del desplazamiento está claramente basado en el lema:

Lema 2.9 *El desplazamiento esperado a causa del punto x es el número de puntos del nodo hermano que contenía x, por la partición hecha en el algoritmo en la definición 4.1*

Es posible ver el caso en que este punto anómalo se pueda coludir con un conjunto de puntos al momento de dibujar un árbol, por lo tanto, en lugar de remover x, se puede remover

un conjunto C con $x \in C$. Análogo a la ecuación 2.1.

$$\mathbb{E}_{T(Z)}[|M(T)|] - \mathbb{E}_{T(Z-C)}[|M(T(Z-C))|] = DISP(C,Z) + \sum_{T} \sum_{y \in C} \mathbb{P}r[T] f(x,Z,T) \tag{2.3}$$

Donde $DISP(C,Z)$ es la noción de desplazamiento extendido a un subárbol, donde $T'' = T(Z-C)$.

$$\sum_{T, y \in Z-C} \mathbb{P}r[T] \left(f(y,Z,T) - f(y,Z-C,T'')\right) \tag{2.4}$$

Teniendo esto en cuenta, se puede observar que el desplazamiento generado por un conjunto C generará que todos los puntos pertenecientes a C tengan el mismo valor de desplazamiento. Por lo tanto una elección natural para determinar C sería $max\ DISP(C, Z/|C|$ sujeto a $x \in C \subset Z$.

Definición 2.10 *Un desplazamiento colusorio de x se denotará como $CoDISP(x, Z, |S|)$ de un punto x se define como:*

$$\mathbb{E}_{S \subset Z, T} \left[max_{x \in X \subset S} \frac{1}{|C|} \sum_{y \in S-C} \left(f(y,S,T) - f(y,S-C,T'')\right) \right] \tag{2.5}$$

Lema 2.11 *$CoDISP(x, Z, |S|)$ puede ser estimado eficientemente.*

Definición 2.12 *Un anómalo corresponde al largo de $CoDISP()$.*

2.1.2. Mantenimiento del bosque con un stream

En esta sección se discuten las formas de mantener un RRCT de manera dinámica. Para eso $RRCF(S)$ es un RRCF sobre la muestra S.

Inserción: Dado un árbol T dibujado de la distribución $RRCF(S)$ y un punto $p \notin S$ genera un T' que proviene de $RRCF(S \cup \{p\})$.

Eliminación: Dado un árbol T dibujado de la distribución $RRCF(S)$ y un punto $p \in S$ genera un T' que proviene de $RRCF(S - \{p\})$.

Eliminación de puntos

Para la eliminación de puntos se sigue el siguiente algoritmo.

Lema 2.13 *Si T es dibujado bajo una distribución de $RRCF(S)$ entonces con algoritmo OlvidarPunto generas a T' el cual es dibujado bajo una distribución de $RRCF(S - \{p\})$*

Algorithm 1 Algoritmo OlvidarPunto

1: Encuentra el nodo v del árbol donde se encuentra el punto anómalo p en T
2: Si u es el nodo hermano de v. Elimina al nodo padre de v y se reemplaza por el nodo u. Es decir, Nodo padre de $v \leftarrow$ Nodo u.
3: Actualiza todos los bounding box del nodo u' hacia arriba - No es necesario para una eliminación pero sí es útil para la inserción
4: Devuelve a T'

Lema 2.14 *La eliminación de un punto se realiza en un tiempo de* $O(\mathrm{d})$ *veces la profundidad del punto p*

Inserción de puntos

El algoritmo de inserción es el siguiente:

Algorithm 2 Algoritmo InsertarPunto

1: Se tiene un set de puntos S' y un árbol $T(S')$. Se desea insertar p y generar el árbol $T'(S' \cup \{p\}$.
2: Si $S' = \Phi$. Se devuelve un nodo que contiene solamente el punto p.
3: En otro caso S' tiene un bounding box $B(S') = [x_1^l, x_1^h] \times [x_2^l, x_2^h] \times [x_3^l, x_3^h] \cdots [x_{\mathrm{d}}^l, x_{\mathrm{d}}^h]$. Donde se tiene que $x_{\mathrm{i}}^l \leq x_{\mathrm{i}}^h$ para todo i.
4: Luego se tiene que para todo i se obtiene un $\hat{x}_{\mathrm{i}}^l = min\{p_{\mathrm{i}}, x_{\mathrm{i}}^l\}$ y $\hat{x}_{\mathrm{i}}^h = min\{x_{\mathrm{i}}^h, p_{\mathrm{i}}\}$.
5: Se escoge un número al azar $r \in [0, \sum_{\mathrm{i}}(x_{\mathrm{i}}^h - x_{\mathrm{i}}^l)]$.
6: Este r corresponde específicamente al corte hecho por la construcción de un $RRCF(S' \cup \{p\})$
7: Actualiza todos los bounding box del nodo u' hacia arriba - No es necesario para una eliminación pero sí es útil para la inserción
8: Devuelve a T'

Lema 2.15 *Si* T *es dibujado bajo una distribución de* $RRCF(S)$ *entonces con algoritmo InsertarPunto genera a* T' *el cual es dibujado bajo una distribución de* $RRCF(S \cup \{p\})$

Y finalmente se define un RRCF y su algoritmo de funcionamiento.

2.2. Consejos de cómo usar RRCF

En esta sección se hablará de consejos de otros trabajos en torno a la forma en que se utiliza RRCF para obtener un mayor provecho.

Se sugiere que en los datos de tiempo se debe utilizar una secuencia de puntos en la que se deben obtener características de esta secuencia, ya sea su promedio, diferencia entre mínimo y máximo, desviación estándar, etc. Como se muestra en la Figura 2.6.

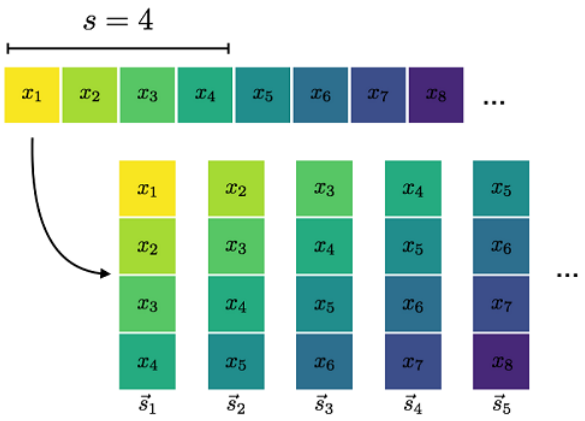


Figura 2.6: Diagrama de experimento. Fuente: [2]

2.3. Métodos de reducción de dimensión

Se recopilaron métodos de reducción de dimensión más conocidos y se investiga cual se adapta a las condiciones planteadas por el alumno:

- *Multidimensional Scaling* (MDS)[9]: Reduce la dimensión intentando preservar la distancia entre cada instancia.[6]. Requiere introducir todos los datos en el modelo. Por lo que no se puede utilizar para un detector de anomalías semi-supervisado.
- *Isomap*: Crea un grafo en que conecta cada instancia a 'vecinos cercanos' (nearest neighbors), luego reduce la dimensión intentando preservar la distancia entre cada instancia[6]. Como trabaja con vecinos cercanos requiere introducir todos los datos en el modelo por lo que no se puede utilizar para un detector de anomalías semi-supervisado.
- *t-Distribuited Stochastic Neighbor Embedding (t-SNE)[10]*: Reduce las dimensiones mientras intenta mantener una instancia similar o cercana a la instancia de partida. Es muy usado para visualización, en particular para *clusters* de grandes dimensiones[6]. Para su correcto funcionamiento requiere introducir todos los datos en el modelo. Por lo que no se puede utilizar para un detector de anomalías semi-supervisado.
- *Uniform Manifold Approximation and Projection (Umap)[11]*: Mediante geometría de Riemannian y topología en algebra, en cuanto a resultados es competitivo con t-SNE, pero para su correcto funcionamiento requiere introducir todos los datos en el modelo. Por lo que no se puede utilizar para un detector de anomalías semi-supervisado.

- *Principal Component Analysis (PCA)*: Uno de los métodos más populares para reducir la dimensión, el cual primero identifica un hiper-plano en donde se maximice la varianza, luego su segundo componente hace el mismo paso, pero es perpendicular al primer hiper-plano y así sucesivamente. Se puede entrenar el modelo utilizando un training set, luego se pueden entrenar los datos anómalos en el mismo modelo. Por lo que cumple los requisitos para la detección de anomalías semi-supervisado.
- *AutoEncoder [4]*: Se utilizan redes neuronales para reducir las dimensiones, por lo que se tiene la capacidad de entrenar el modelo utilizando un training set, luego se pueden entrenar los datos anómalos en el mismo modelo. Por lo que cumple los requisitos para la detección de anomalías semi-supervisado.

2.4. Tres distintos escenarios

Dado lo hablado en la Sección 2.3, sólo se podrá llevar a cabo tres escenarios distintos para poder responder la hipótesis, que serán nombrados como: Sin reducción de dimensión, Reducción con PCA y Reducción con AutoEncoder.

2.4.1. Sin reducción de dimensión

Como se observa en la Figura 2.7 se utiliza un set de datos sin anomalías para entrenar al modelo de RCCF. Luego se ingresa la base de datos que se quiere clasificar entre dato anómalo o dato no anómalo.

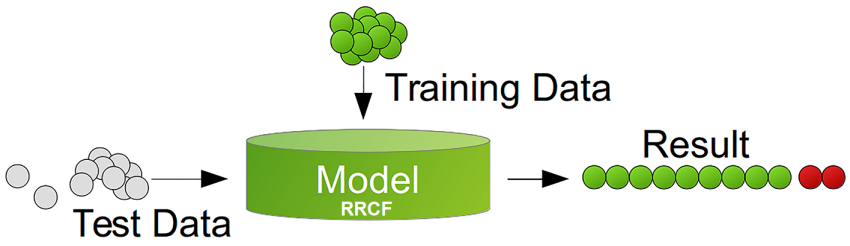


Figura 2.7: Modelo de entrenamiento sin reducción de dimensión. Fuente: Adaptación de [8].

2.4.2. Sin reducción de dimensión

Como se observa en la Figura **??** se utiliza un set de datos sin anomalías para entrenar al modelo de PCA y luego el de RCCF. Luego se ingresan la base de datos que se quiere clasificar entre dato anómalo o dato no anómalo.

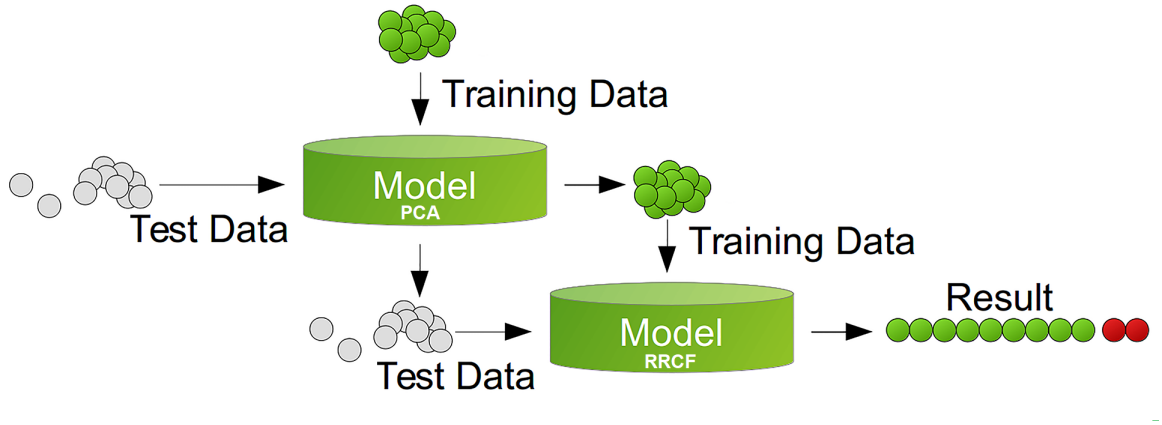


Figura 2.8: Modelo de entrenamiento con PCA. Fuente: Adaptación de [8].

2.4.3. Sin reducción de dimensión

Como se observa en la Figura 2.9 se utiliza un set de datos sin anomalías para entrenar al modelo de PCA y luego el de RCCF. Luego se ingresan la base de datos que se quiere clasificar entre dato anómalo o dato no anómalo.

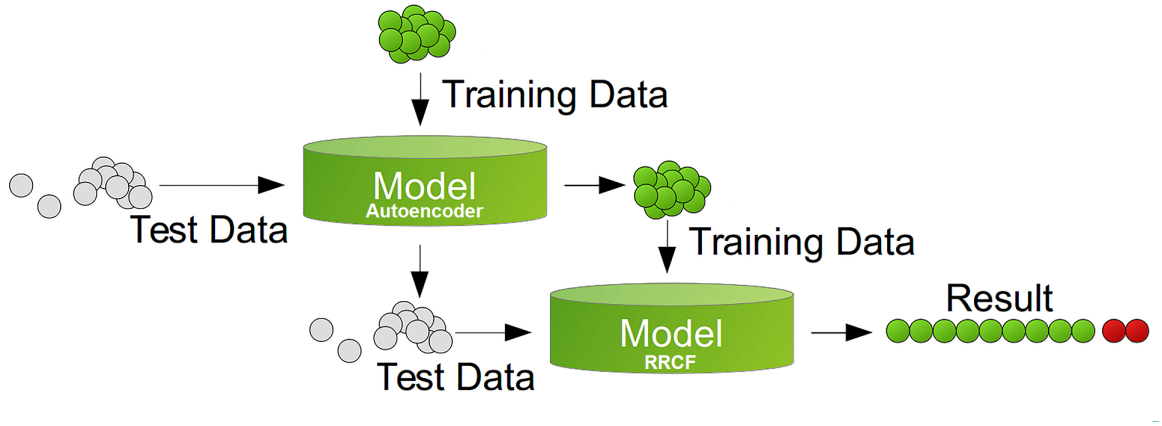


Figura 2.9: Modelo de entrenamiento con AutoEncoder. Fuente: Adaptación de [8].

Capítulo 3

Experimento de barras

En esta sección se hablará del experimento realizado para obtener los datos de vibración que serán utilizados para el Robust Random Cut Forest.

3.1. Barras de acero

Se tienen cuatro barras de acero A270ES de medidas de largo 850 [mm], ancho 10 [mm] y alto 25 [mm][12]. A las cuales se les harán las siguientes modificaciones:

- **Barra Acero 1 [Healthy]:** Sus dimensiones se mantienen tal cual, a esta barra se le denominará "Healthy".
- **Barra Acero 2 [h5mm]:** Sus dimensiones se mantienen no obstante, se le realiza un daño intencionado a 250 mm de un extremo, este daño consiste un corte de profundidad de 5 [mm] por 1 [mm] ancho por el largo de los 25 [mm], a esta barra se le denominará como "h5mm".
- **Barra Acero 3 [h2mm]:** Sus dimensiones se mantienen no obstante, se le realiza un daño intencionado a 250 mm de un extremo, este daño consiste un corte de profundidad de 2 [mm] por 1 [mm] ancho por el largo de los 25 [mm], a esta barra se le denominará como "h2mm".
- **Barra Acero 4 [h1mm]:** Sus dimensiones se mantienen no obstante, se le realiza un daño intencionado a 250 mm de un extremo, este daño consiste un corte de profundidad de 1 [mm] por 1 [mm] ancho por el largo de los 25 [mm], a esta barra se le denominará como "h1mm".

3.2. Montaje del experimento

El montaje del experimento consiste en la siguiente instalación.

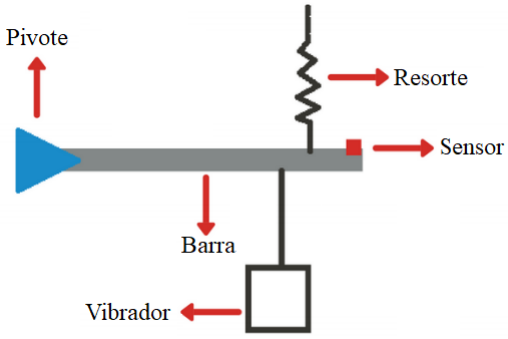


Figura 3.1: Diagrama de experimento. Fuente: Elaboración propia.

Como se puede ver en la Figura 3.1 en un extremo de la barra se monta a un pivote, por el otro extremo se acomoda el sensor en el borde, a 4 [cm] aproximadamente del sensor PCB modelo 333B32 se instala el resorte y el vibrador SINOCERA modelo JZK-10. El vibrador SINOCERA modelo JZK-10 reproducirá el ruido blanco que fue producido por el generador de señal SINOCERA modelo YE1311 y amplificado por el amplificador de señal SINOCERA modelo YES871A. La señal generada sin amplificar se observa mediante el osciloscopio Tektronix modelo TDS 210, para ver el funcionamiento correcto del generador de señal SINOCERA modelo YE1311. Finalmente, la recibida por el sensor PCB modelo 333B32 se guarda en una tarjeta de adquisición y la información guardada en un computador. Finalmente queda el ensayo experimental queda como en la Figura 3.2.

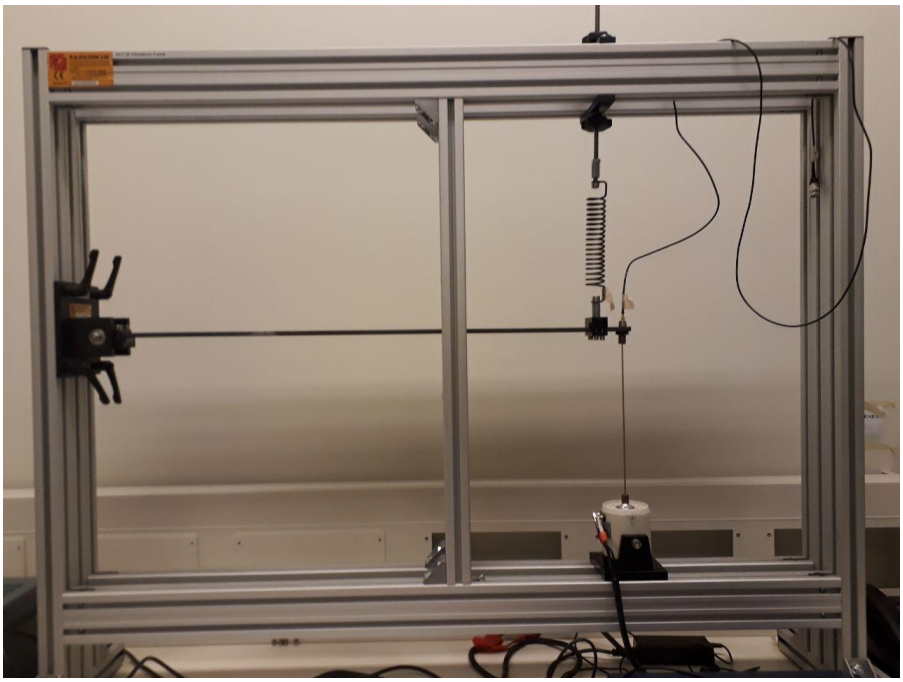


Figura 3.2: Ensayo Experimental. Fuente: Elaboración propia.

3.3. Extracción de datos

Para evitar error de montaje, la medición de cada barra se repite, para así poder capturar las vibraciones de cada barra. Por lo que se realizó una iteración experimento de barra de la siguiente manera:

Tabla 3.1: Tabla resumen de extracción de datos.

Barra	Orden de montaje	Tiempo	Número de puntos
3*Healthy	Primero	4 Minutos 15 Segundos	1,048,575
	Quinto	4 Minutos 15 Segundos	1,048,575
	Noveno	4 Minutos 15 Segundos	1,048,575
2*h5mm	Segundo	4 Minutos 15 Segundos	1,048,575
	Sexto	4 Minutos 15 Segundos	1,048,575
2*h2mm	Tercero	4 Minutos 15 Segundos	1,048,575
	Séptimo	4 Minutos 15 Segundos	1,048,575
2*h1mm	Cuarto	4 Minutos 15 Segundos	1,048,575
	Octavo	4 Minutos 15 Segundos	1,048,575

Como se puede observar en la Tabla 3.1, el orden en que se hizo la toma de datos fue healthy, h5mm, h2mm, h1mm, healthy, h5mm, h2mm, h1mm y healthy. Además, se puede ver el tiempo en que se toman datos en cada montaje y finalmente, el número de puntos que se obtienen de cada montaje. La última columna es relevante pues quiere decir que la frecuencia de muestreo en que se hizo el experimento es de 4,096[hz], es decir que toma 4,096 puntos por cada segundo de experimento.

Mediante el programa FlexLogger, como se puede observar en la Figura 3.3 se puede obtener una visualización previa de los datos obtenidos por el sensor, además de registrar que los datos son de aceleración expresados en milivoltios $[mV]$. Tras exportar los datos experimentales se puede trabajar con Python.

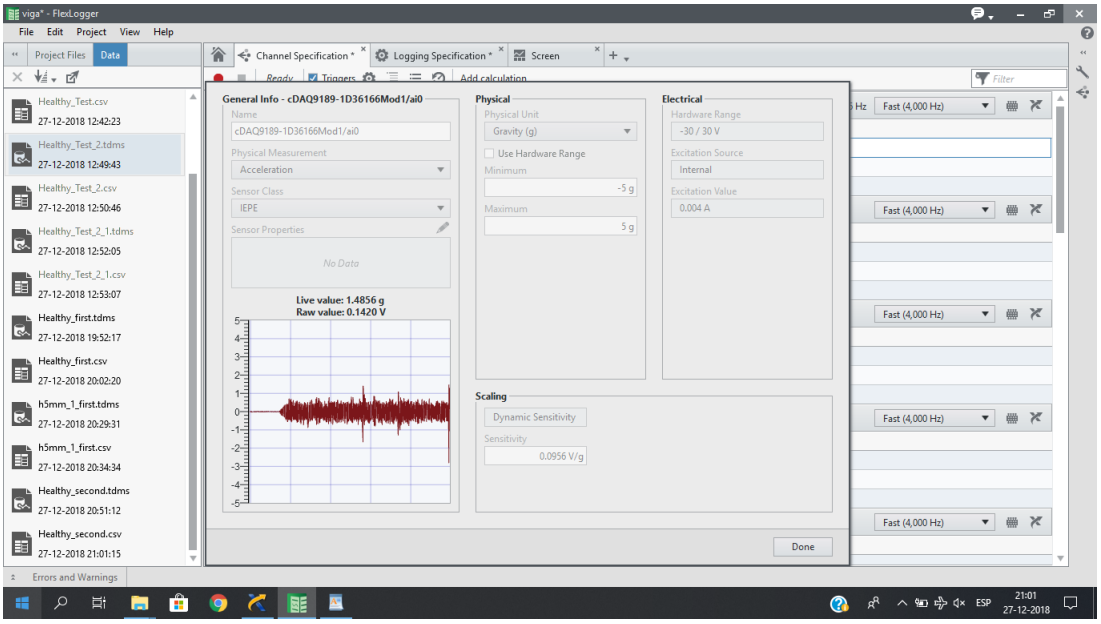


Figura 3.3: Software FlexLogger. Fuente: Elaboración propia.

Capítulo 4

Desarrollo

En este capítulo se hablará de la metodología de trabajo que permite llegar a los resultados. Para poder desarrollar la sección correctamente se recurre a los siguientes pasos.

- Prueba de Robust Random Cut Forest.
- Mejora en Robust Random Cut Forest.
- Visualización de datos.
- Entrenando para detección de anomalías.

4.1. Prueba de Robust Random Cut Forest

Para la primera parte se debe ver como funciona esquemáticamente RRCF por consiguiente, se crea una base de datos de 300 puntos. Esta base de datos consiste en 300 puntos que forman parte de un círculo que se encuentra entre el 0 y el 1, en el eje X (dimensión 0) y en el eje Y (dimensión 1), excepto un punto que será considerado el punto número 150, este punto número 150 se encuentra notoriamente fuera del rango 0 y 1 como se puede ver en la Figura 4.1. Por lo que el objetivo de esta sección es ver como el algoritmo planteado en los antecedentes llega a su fin de detectar el punto anómalo, por lo mismo se seguirán paso a paso las definiciones entregadas por el RRCF para una profundidad o altura de árbol de 3.

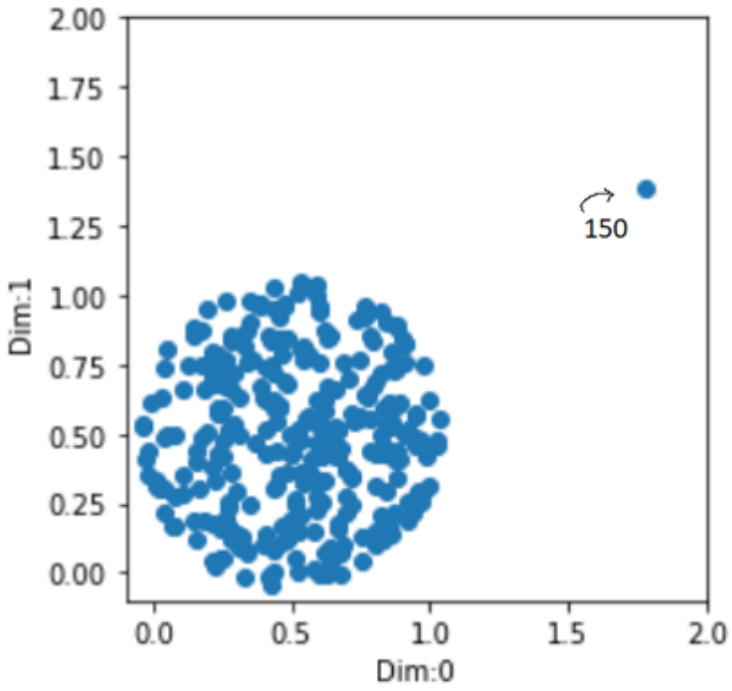


Figura 4.1: Base de datos: Círculo. Fuente: Elaboración propia.

En el algoritmo por Definición 4.1, se debe elegir una dimensión de manera aleatoria y proporcional a $\frac{l_i}{\sum_j l_j}$, y la definición de l_i es $max_{x \in S} x_i - min_{x \in S} x_i$. Entonces se debe observar la base de datos, si se observan los límites de cada dimensión de la muestra de datos como se observa en Figura 4.2 se puede ver que, aunque la base de datos nominal va entre 0 y 1, el punto anómalo se ubica en (1.8, 1.5) lo que da mayor probabilidad de que la dimensión 0 sea seleccionada.

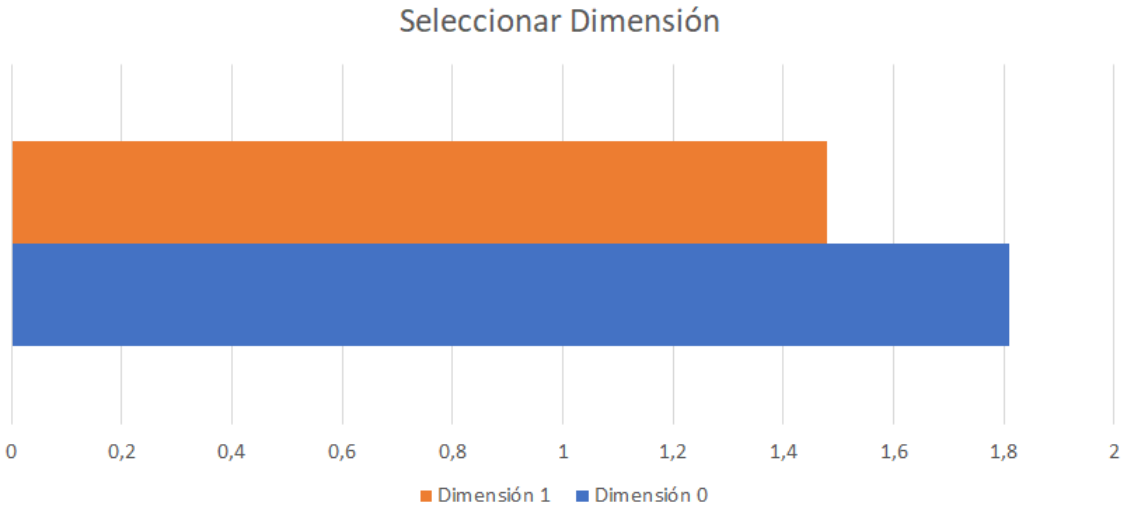


Figura 4.2: Límites de cada dimensión. Fuente: Elaboración propia.

La proporción $\frac{l_i}{\sum_j l_j}$ se puede ver reflejada en la Figura 4.3. La probabilidad que tiene la dimensión 0 de ser seleccionada es $\frac{1{,}8}{1{,}8+1{,}5} = 54{,}54\,\%$

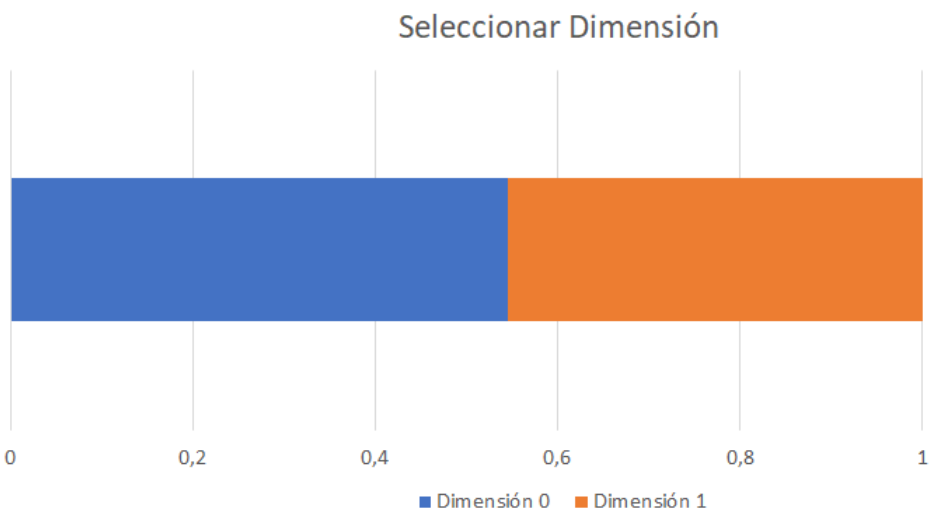


Figura 4.3: Dimensiones proporcionadas en base a los tamaños de la misma. Fuente: Elaboración propia.

Siguiendo por la definición de RRCF, mediante una probabilidad uniforme en base a la proporción anterior, se debe seleccionar una dimensión, según el ejemplo se obtuvo el caso de la dimensión 0. Como se muestra en la Figura 4.4.

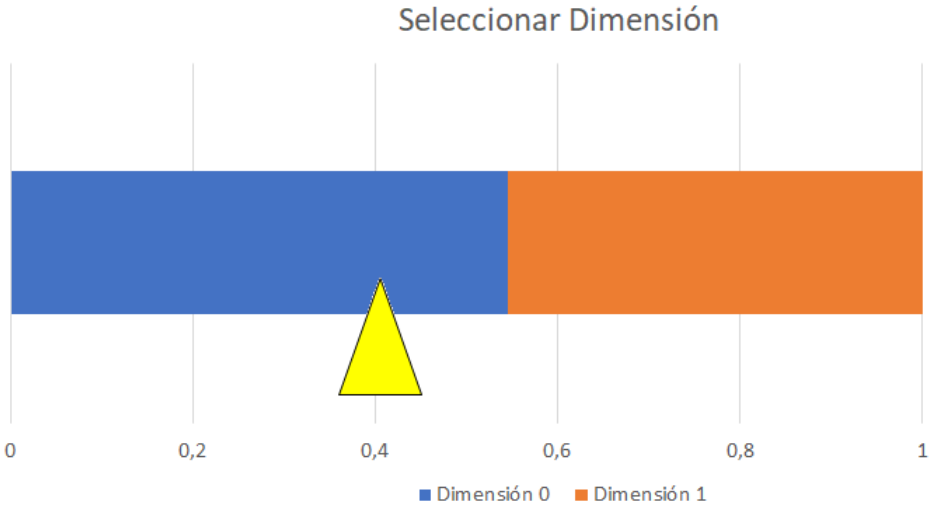


Figura 4.4: Selección de dimensión. Fuente: Elaboración propia.

Ya teniendo la dimensión 0 seleccionada, se sigue con el paso 2 en donde se elige un valor de manera aleatoria en una distribución uniforme entre los rangos de la dimensión, es decir, $X_i \sim \text{Uniforme}[\min_{x \in S} x_i, \max_{x \in S} x_i]$. El valor obtenido se considera el valor de corte a la base de datos. La selección de este valor se puede ver reflejado en la Figura 4.5

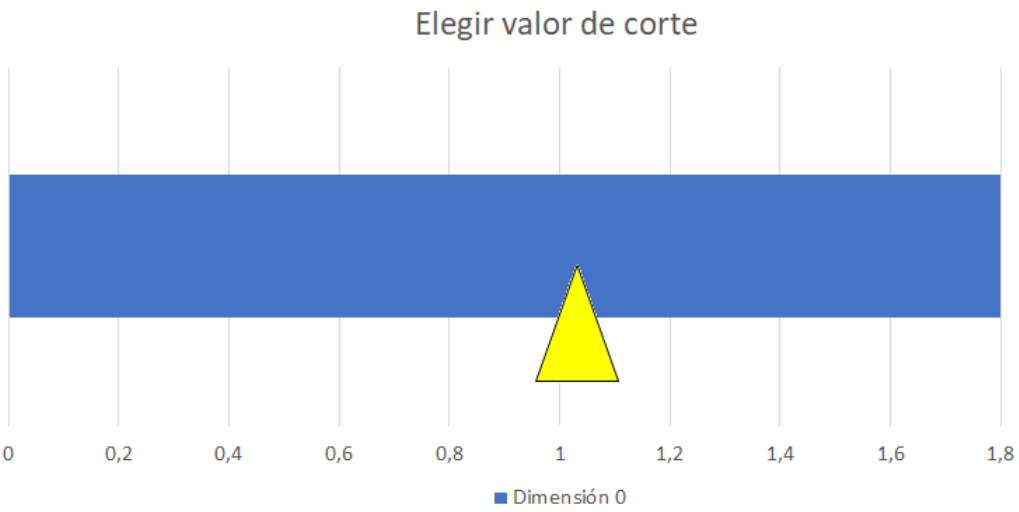
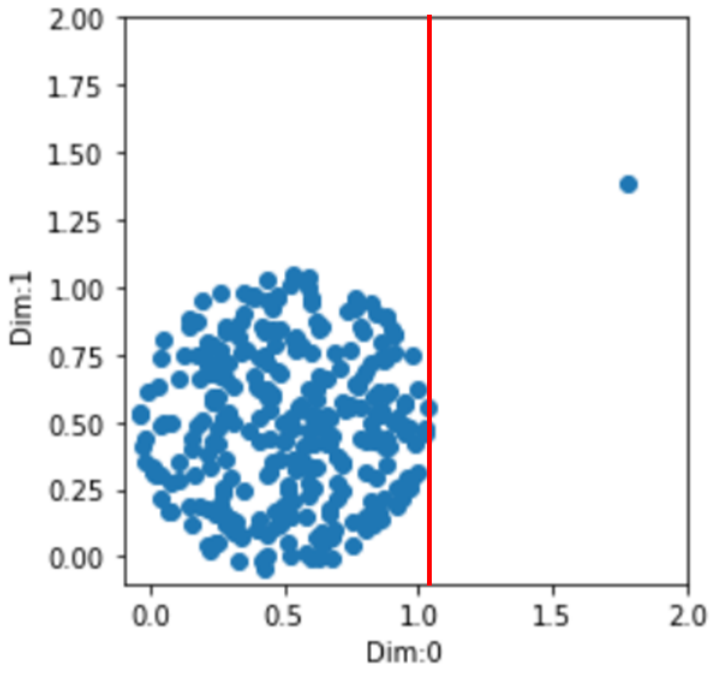


Figura 4.5: Selección de dimensión. Fuente: Elaboración propia.

En el ejemplo se obtuvo un valor de corte de 1,01. Para poder representar esto gráficamente en la muestra de datos se realizó la siguiente Figura 4.6. Que quiere mostrar un corte de valor 1,01 en la dimensión 0.



1.-	**0**	**(Dimensión)**
2.-	**1,01**	**(Corte)**

Figura 4.6: Primer corte. Fuente: Elaboración propia.

También se puede mostrar este corte mediante un árbol de decisión/binario, el cual se puede representar como en la Figura 4.7

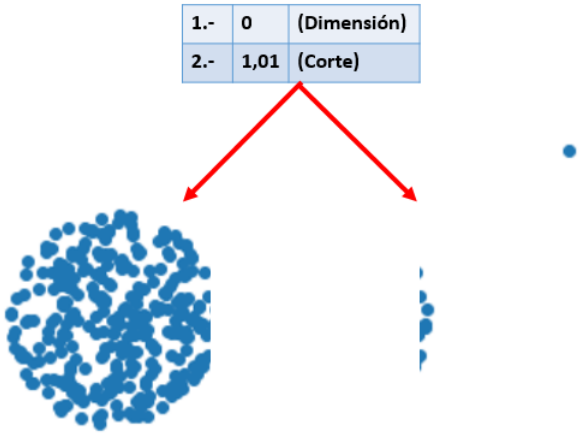


Figura 4.7: Representación de un árbol del primer corte. Fuente: Elaboración propia.

Luego en el paso 3 de la Definición 4.1 se deben realizar los pasos 1 y 2 recursivamente hasta llegar a la profundidad máxima deseada. En el siguiente corte se tiene un subconjunto naranjo del lado izquierdo y uno verde del lado derecho, al subconjunto del lado izquierdo se selecciona un corte de valor 0,4 en la dimensión 1 y al subconjunto del lado derecho se selecciona un corte de valor 1,3 en la dimensión 0.

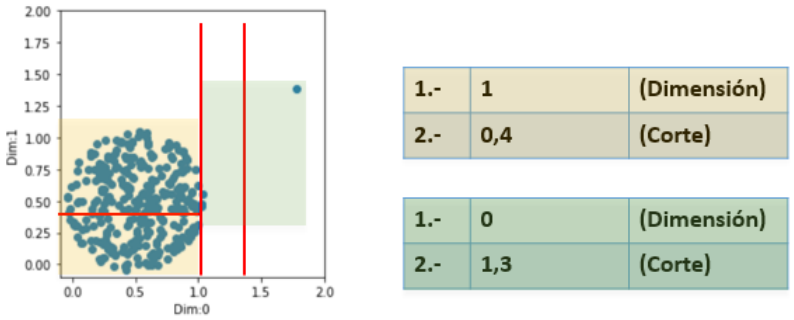


Figura 4.8: Segundo corte. Fuente: Elaboración propia.

Si se arma el árbol de decisión/binario, quedaría como se muestra en la Figura 4.10.

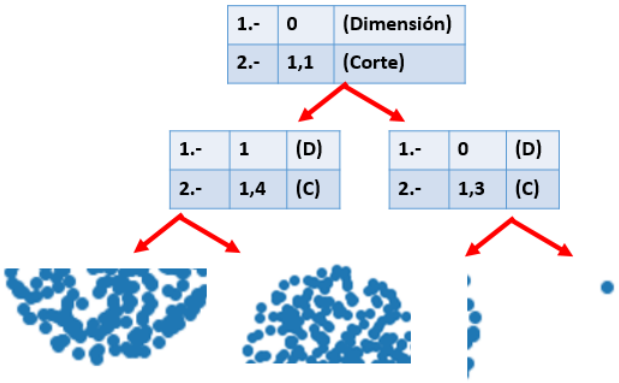


Figura 4.9: Árbol de segundo corte. Fuente: Elaboración propia.

Luego se realiza el mismo procedimiento para cada subconjunto creado, para el lado naranjo se obtuvo 0,75 en la dimensión 0 y en el subconjunto del lado derecho se selecciona un corte de valor 0,44 en la dimensión 0. Como se muestra en la Figura 4.10.

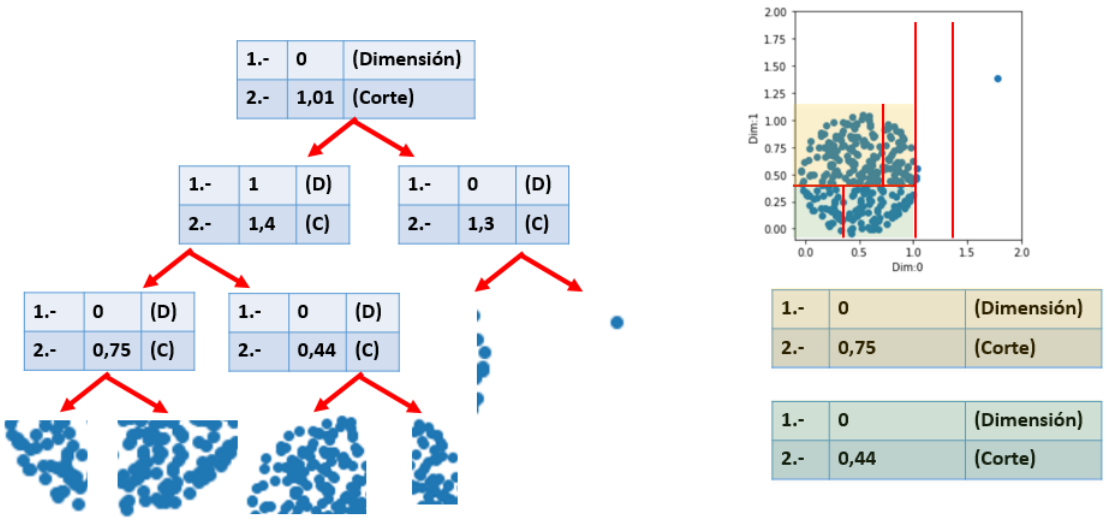


Figura 4.10: Árbol de tercer corte. Fuente: Elaboración propia.

En el cuarto corte, observa que el punto anómalo ya se encuentra apartado, pero aun así hay que realizar un corte en el nodo hermano como se muestra en el que salió el valor de corte 0,51 en la dimensión 1. Como se muestra en la Figura 4.11.

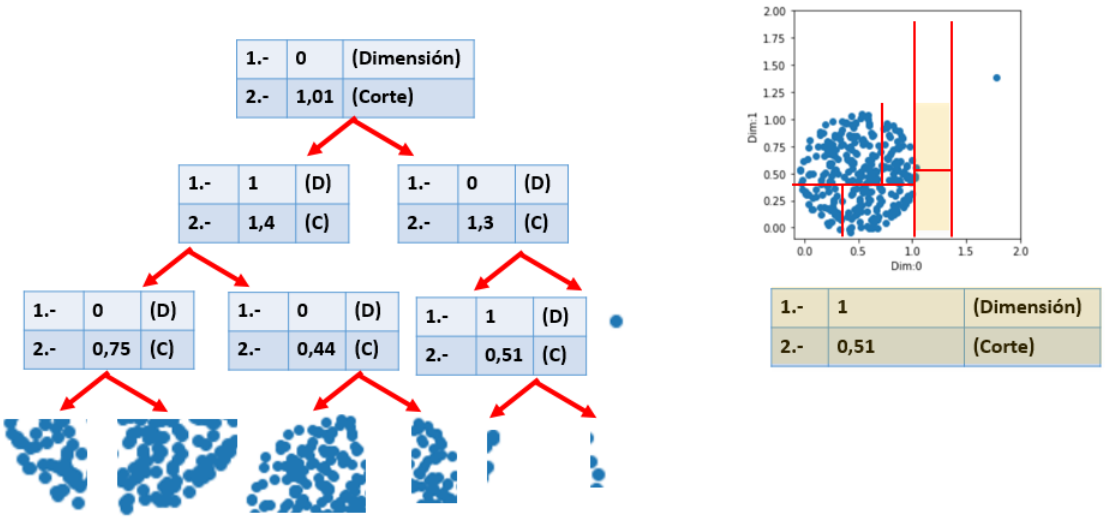


Figura 4.11: Árbol de cuarto corte. Fuente: Elaboración propia.

Finalmente, se obtiene el primer RRCT de profundidad tres. Hay que recordar que para este algoritmo se hace un conjunto de árboles, por lo que se repite el proceso hasta llegar a la cantidad de árboles deseados.

Ya armados los RRCT se sabe que un punto anómalo se define según el largo de $CoDISP(x, Z, |S|)$ por la Definición 2.5, esta definición depende de $f(y, S, T)$ que no es entregada por el paper, pero guía a una intuición de qué función puede ser. Para poder llegar a esta función se pueden considerar las siguientes variables:

- **Nodo Supervisado:** Entrega el número de datos que contiene el nodo en que se encuentra el punto que se está estudiando, éste sería el nodo en que se supone que se encuentra la anomalía, esta variable será denotada como n_s.
- **Nodo Hermano:** Entrega el número de datos que contiene el nodo hermano que se encuentra bajo estudio (n_s), esta variable será denotada como n_h.
- **Nodo Padre:** Entrega el número de datos que contiene el nodo padre que se encuentra bajo estudio (n_s), esta variable será denotada como n_p, observar que el número de datos que contiene el nodo padre equivale a $n_p = n_s + n_h$.
- **Nodo raíz o número total de datos:** Entrega el número de datos que contiene el nodo raíz del árbol, es decir entrega el número total de datos que se encuentran en el árbol, esta variable será denotada como n_t.

Ya definidas estas variables se proponen dos definiciones para la función $f(y, S, T)$:

- **Anomalías por diferencias:**

$$\left(\frac{n_h - n_s}{n_t}\right) \cdot \left(\frac{n_p}{n_t}\right) \tag{4.1}$$

Con el primer término se busca una relación entre el nodo supervisado y nodo hermano, donde entre más grande sea la diferencia de nodos mayor será la cantidad de nodos que serán desplazados si falta el nodo supervisado. Y con el segundo término lo que se

quiere es ponderar el nivel de profundidad de los nodos por si se da el caso de que las diferencias de número de puntos sean iguales.

- **Anomalía por proporción:**

$$\left(\frac{n_h}{n_s}\right) \cdot \left(\frac{1}{n_t}\right) \tag{4.2}$$

En este caso hay que centrarse en la diferencia proporcional entre nodos hermanos y finalmente, se pondera por el número total de datos. Se puede observar que si un punto anómalo es fácilmente encontrado en un árbol el valor de $f(y, S, T)$ la proporción $\frac{n_h}{n_s}$ será alta, por lo que bonificará. Es decir, esta forma de calcular $f(y, S, T)$ no depende directamente de la altura en que se encuentra el nodo.

Luego, para el caso de la Figura 4.1, se pueden obtener los siguientes resultados:

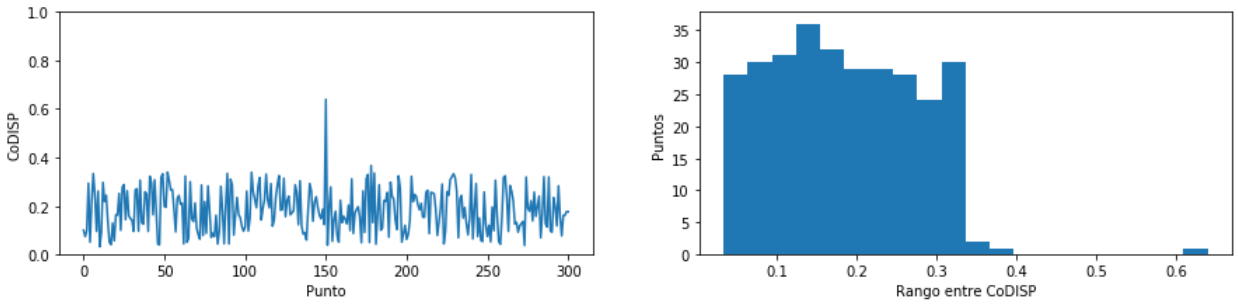


Figura 4.12: Anomalía por diferencia. Fuente: Elaboración propia.

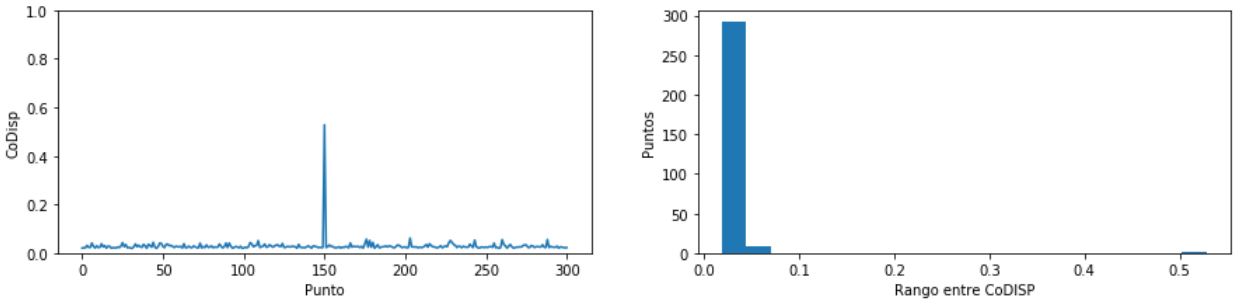


Figura 4.13: Anomalía por proporción. Fuente: Elaboración propia.

En ambas Figuras 4.12 y 4.13, se tiene el gráfico izquierdo que entrega los resultados de $CoDISP$ para cada punto generado en la base de datos y en el lado derecho se hace una distribución de los puntos según el rango que corresponde su $CoDISP$.
Se puede observar que ambos, en el punto 150, tienen un valor $CoDISP$ de 0.62 y 0.51 para anomalía por diferencia y anomalía por proporción, respectivamente. Aunque en anomalía por diferencia se tiene un $CoDISP$ más alto, también los valores no anómalos se encuentran en un rango de $[0, 0{,}4]$ lo que se diferencia de anomalía por proporción. Sus valores no anómalos se encuentran en un rango de $[0, 0{,}08]$ por lo que se puede observar que se tiene una detección de anomalías más controlada, que se puede dar que no depende de la altura directa en

que se encuentra el punto sino del desplazamiento que genera la extracción de un punto. Estos resultados se siguen repitiendo para ejemplos que se presentarán posteriormente, pero desde ahora adelanta que se utiliza la función "Anomalía por proporción" para identificar las anomalías.

4.2. Mejora en Robust Random Cut Forest

Posteriormente se quiere ver si es posible mejorar el RRCF. Para lo que se genera una base de datos más complicada, en la que se propone generar una elipse inclinada con un punto anómalo aparentemente dentro del bounding box inicial del RRCT.

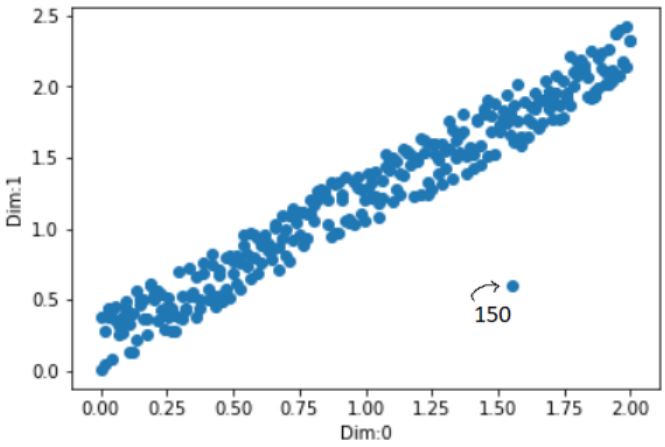


Figura 4.14: Base de datos: Elipse inclinada. Fuente: Elaboración propia.

Un bounding box inicial de RRCT refiere a que los valores nominales o los valores no anómalos de la elipse se encuentran dentro de los rangos $[0, 2]$ para la dimensión 0 y $[0, 2{,}5]$ para la dimensión 1, mientras el punto 150 se encuentra en la posición $(1{,}55, 0{,}6)$ lo que en ambos valores lo hace inicialmente pertenecer al rango de los valores nominales. Pero posteriormente, después de un corte este bounding box de rangos nominales se puede acortar y finalmente dejar al punto anómalo fuera del rango nominal. Por ejemplo, suponiendo que el primer corte se realiza en el valor 1.25 en la dimensión 0. Luego el bounding box de los datos no anómalos del nodo en que pertenece el punto 150 se encuentran dentro de los rangos $[1, 25, 2]$ para la dimensión 0 y $[1{,}1, 2{,}5]$ para la dimensión 1 lo que haría que posteriormente el punto $(1{,}55, 0{,}6)$ se encuentre como un punto anómalo.
Por lo que haciendo correr el RRCF se llega al siguiente resultado: Se puede ver que el punto anómalo 150 es reconocible por lo anteriormente explicado además, que la función $f(y, S, T)$ no depende directamente de la profundidad en que se encuentra la anomalía, pero el análisis anteriormente planteado sugiere la existencia de una pequeña mejora al momento antes de realizar un corte. Para esto se propone utilizar Principal Component Analysis (PCA) pero no para reducir dimensión, sino para realizar una rotación, que es una de las características de PCA.
Como se ha observado, los cortes se realizan perpendicular a los ejes que se definen en cada

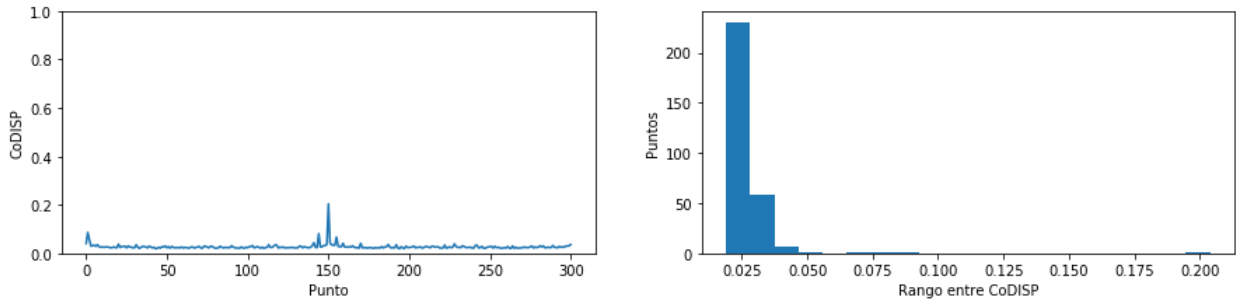


Figura 4.15: Detección de anomalía en elipse inclinada. Fuente: Elaboración propia.

dimensión, por lo que al realizar una rotación de sus componentes principales lo ayudará a detectar el punto anómalo.

Ahora una explicación gráfica de lo que sucede al aplicar PCA. Como se puede ver en la Figura 4.16 se destaca cual es la componente principal. Luego con PCA se puede ver cuales son un mejor componente principal de manera que existe menor varianza entre sus ejes y los datos. Este nuevo componente principal se destaca en naranjo en la Figura 4.17. Finalmente la muestra de datos se rota y se definen los nuevos componentes principales como eje.

Observación Se debe ver que al realizar esta rotación los bounding box de los datos varían, que es lo que se pretende hacer para facilitar la búsqueda de puntos anómalos.

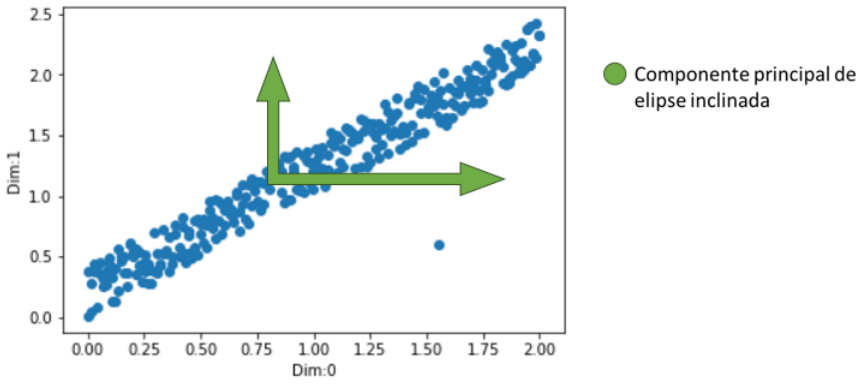


Figura 4.16: Componente principal en elipse inclinada. Fuente: Elaboración propia.

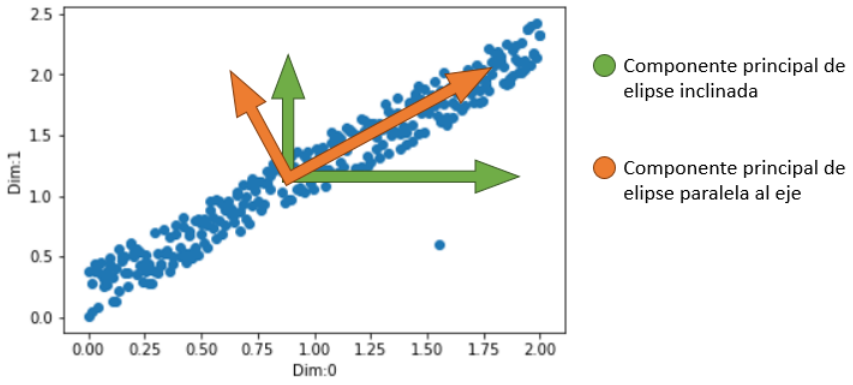


Figura 4.17: Nueva componente principal en elipse inclinada. Fuente: Elaboración propia.

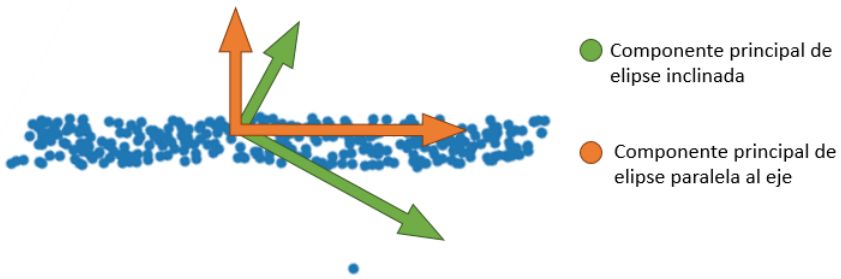


Figura 4.18: Componente principal en elipse ajustada. Fuente: Elaboración propia.

Al implementar este paso a la Definición 4.1, sería antes del paso 1 y quedaría como:

Definición 4.1 (Robust Random Cut Forest con rotación) *Un árbol de corte aleatorio robusto con PCA, Robust Random Cut Tree con rotación (RRCT-Rot) de un conjunto de datos S es generado por los siguientes pasos:*

1. *Se realiza un PCA rotacional a la muestra de datos S.*
2. *Se elige una dimensión de manera aleatoria proporcional a $\frac{l_i}{\sum_j l_j}$, donde l_i es la distancia de los dos puntos más lejanos de la dimensión* i*, es decir, $l_i = max_{x \in S} x_i - min_{x \in S} x_i$.*
3. *Se elige un $X_i \sim Uniforme[min_{x \in S} x_i, max_{x \in S} x_i]$.*
4. *Define $S_1 = \{x | x \in S, x_i \leq X_i\}$ y $S_2 = S - S_1$ y luego aplicar esta definición recursivamente en S_1 y S_2*

Finalmente un RRCF con rotación (RRCF-Rot) es el conjunto de RRCT-Rot.

La implementación anterior fue propuesta y realizada por Mohammad H. Pishahang.

Si se utiliza la nueva definición que se entregó de RRCF-Rot a la base de datos de elipse inclinada, es decir a la Figura 4.14, entrega la Figura 4.19. Se puede ver que la detección del punto anómalo número 150 se realiza con mayor facilidad que sin PCA esto se puede ver directamente en la Figura 4.20.

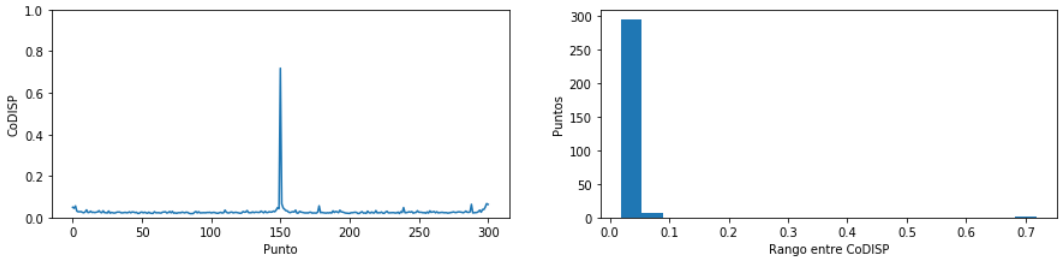


Figura 4.19: Detección de anomalía en elipse inclinada con RRCF-Rot. Fuente: Elaboración propia.

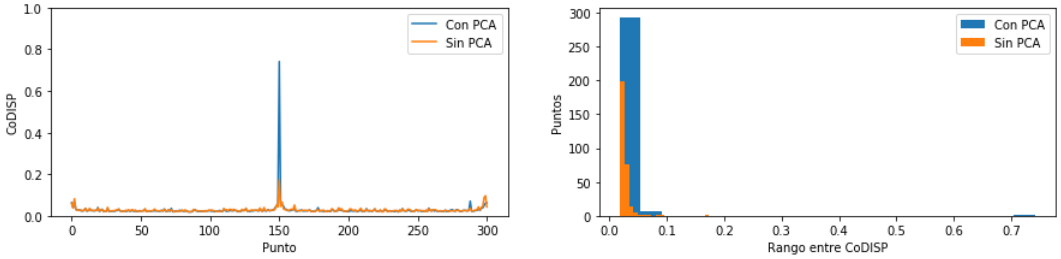


Figura 4.20: Comparación de RRCF con RRCF-Rot en detección de anomalía en elipse inclinada. Fuente: Elaboración propia.

Observación Aunque en ambos casos el punto anómalo es detectado, esta mejora servirá en casos en que la muestra de datos sea más compleja, como se verá a continuación.

Ahora se realizarán unas bases de datos donde la ubicación del punto anómalo puede ser interesante de encontrar. Base de datos de V, consiste en una letra V y un punto anómalo.

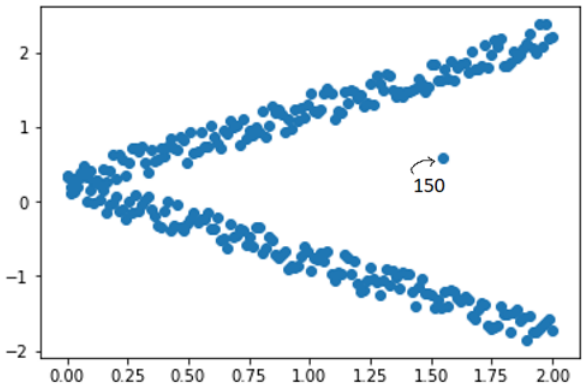


Figura 4.21: Base de datos: V. Fuente: Elaboración propia.

Resultados comparativos:

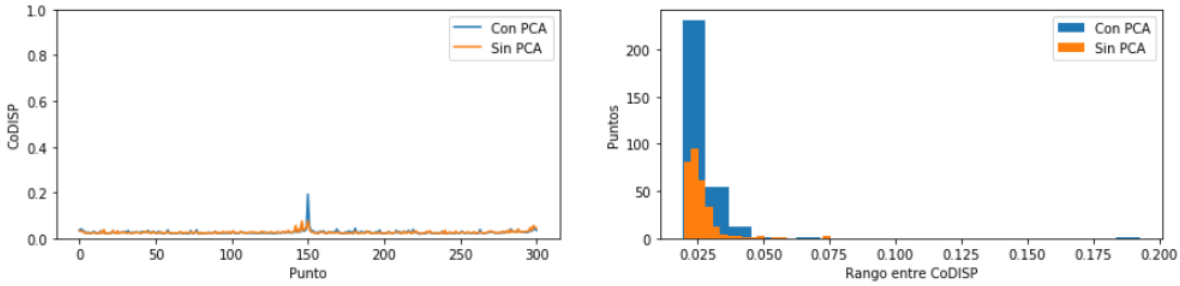


Figura 4.22: Comparación de RRCF con RRCF-Rot en detección de anomalía en V. Fuente: Elaboración propia.

Base de datos de circunferencia con anomalía en el centro, consiste en una circunferencia y un punto anómalo en el centro. Resultados comparativos:

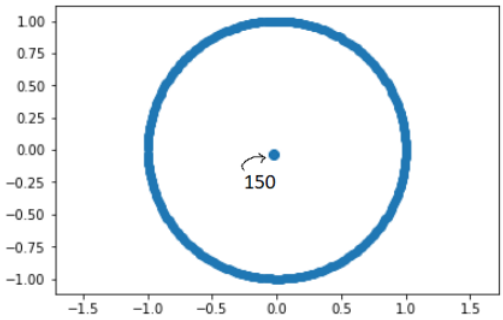


Figura 4.23: Base de datos: circunferencia con anomalía en el centro. Fuente: Elaboración propia.

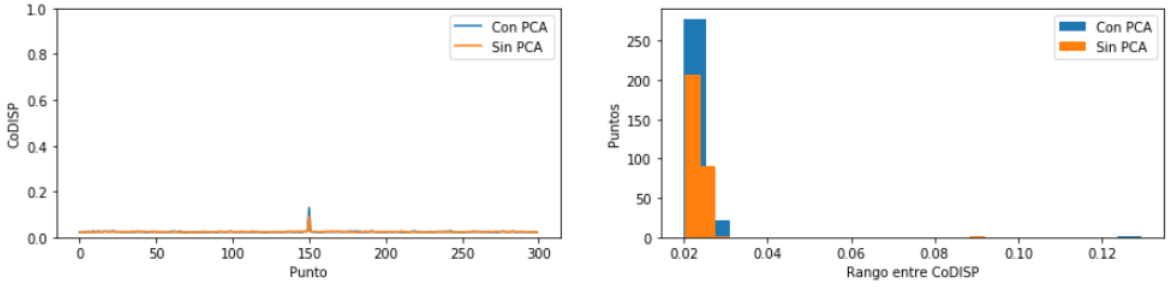


Figura 4.24: Comparación de RRCF con RRCF-Rot en detección de anomalía en circunferencia con anomalía en el centro. Fuente: Elaboración propia.

Finalmente cuando ya se obtiene el puntaje de CoDISP de cada punto, se debe definir un *threshold*, también conocido como un límite, en el cual se llega a la conclusión de considerar que el 98 % de los puntos ingresados van a quedar dentro de este límite, y dejando que el 2 % se quedan anómalos, de esta manera se puede definir un *BoundingBox*. Ahora si se utiliza en la base de dato del círculo, en la Figura 4.25 se puede observar que los puntos amarillos son los considerados anómalos.

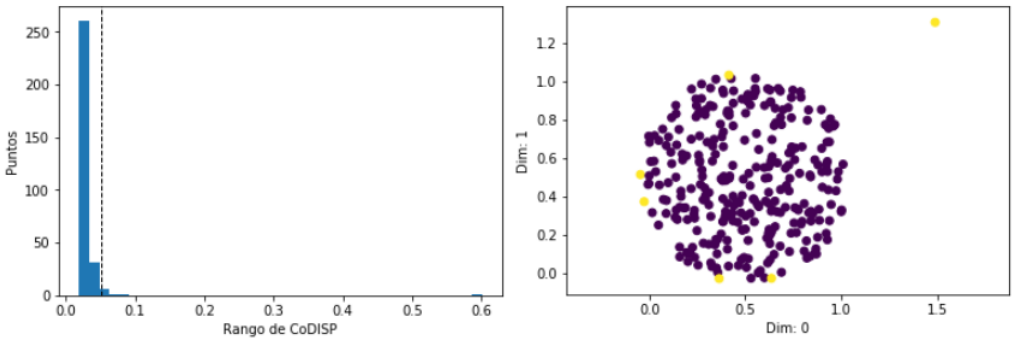


Figura 4.25: Data de base de un círculo, señalando los puntos anómalos en amarillo. Fuente: Elaboración propia.

4.3. Preprocesamiento de datos.

Ya estudiado y programado RRCF-Rot, se observan los datos que son recibidos por la tarjeta de adquisición. Como se puede ver en Figura 4.26, se tiene una grabación constante de lo que parece ruido blanco.

Observación Antes de poder trabajar con RRCF-Rot con Stream se sugiere, por parte de la bibliografía, generar un nuevo set de datos en donde un dato i dependa de cuatro datos siguientes $i+1, i+2, i+3, i+4$, es decir, que ahora el dato i puede ser un dato en que se tiene en cada dimensión promedio, la desviación estándar y la diferencia del mínimo y el máximo de los $i, i+1, i+2, i+3, i+4$.

En este caso experimental, se tienen datos de vibración en los que trabajar con promedios, desviación estándar en ruido blanco se pierde significado, no obstante se tiene la posibilidad de poder trabajar mediante la transformada de Fourier de los datos.

Ya que la tarjeta de adquisición fue programada para trabajar en 4096 [hz], se obtiene la transformada de Fourier de 4096 puntos, es decir, la transformada de Fourier de un segundo de grabación.
Esto genera un dato de 4096 dimensiones para la RRCF recordar que según Tabla **??** por cada montaje realizado se obtienen 1,048,575 puntos, entonces por cada montaje se pueden tener aproximadamente 255 datos de transformada de Fourier.

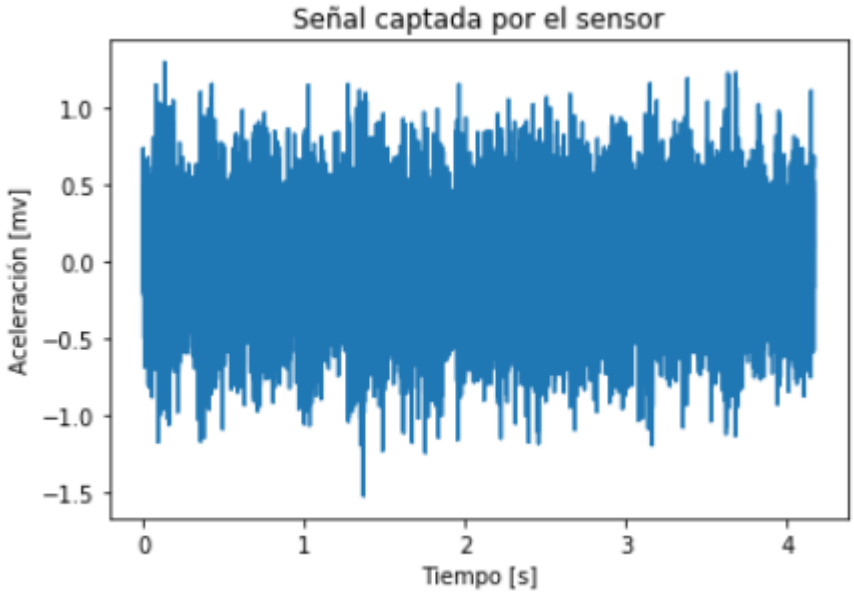


Figura 4.26: Datos crudos. Fuente: Elaboración propia.

Entonces la nueva tabla de resumen de datos es la siguiente:

Tabla 4.1: Tabla Resumen de datos.

Barra	Orden de montaje	Tiempo	Número de puntos	Número de datos
Healthy	Primero	4 Minutos 15 Segundos	1,048,575	767
	Quinto	4 Minutos 15 Segundos	1,048,575	
	Noveno	4 Minutos 15 Segundos	1,048,575	
h5mm	Segundo	4 Minutos 15 Segundos	1,048,575	511
	Sexto	4 Minutos 15 Segundos	1,048,575	
h2mm	Tercero	4 Minutos 15 Segundos	1,048,575	511
	Séptimo	4 Minutos 15 Segundos	1,048,575	
h1mm	Cuarto	4 Minutos 15 Segundos	1,048,575	511
	Octavo	4 Minutos 15 Segundos	1,048,575	

Notar que en la tabla se agregó la columna 'Número de datos' esto es un dato interesante ya que de los datos de Healthy para la detección de datos se utiliza un Test_size=0.2 es decir 613 datos son para entrenar a la redes neuronales y 154 datos para el RRCF-Rot. En la Figura 4.28 se puede ver como se resultan las transformada de Fourier de los datos Healthy.

Observación Sabiendo que los datos tienen 4,096 dimensiones, se pueden tomar unas 6 dimensiones aleatorias del conjunto de datos y poder visualizar. Tanto cuando eran datos crudos como son con trasformada de Fourier. Estas imágenes pueden ser útiles para ver como mejora la reducción de dimensiones en cada caso, además de poder ver que desde un comienzo la transformada de Fourier puede ayudar un poco en agrupar los datos. Se puede observar que la transformada de Fourier es simétrica, por lo que de 4,096 puntos se utilizan 2,048 puntos, reduciendo las dimensiones a la mitad simplemente haciendo la transformada de Fourier.

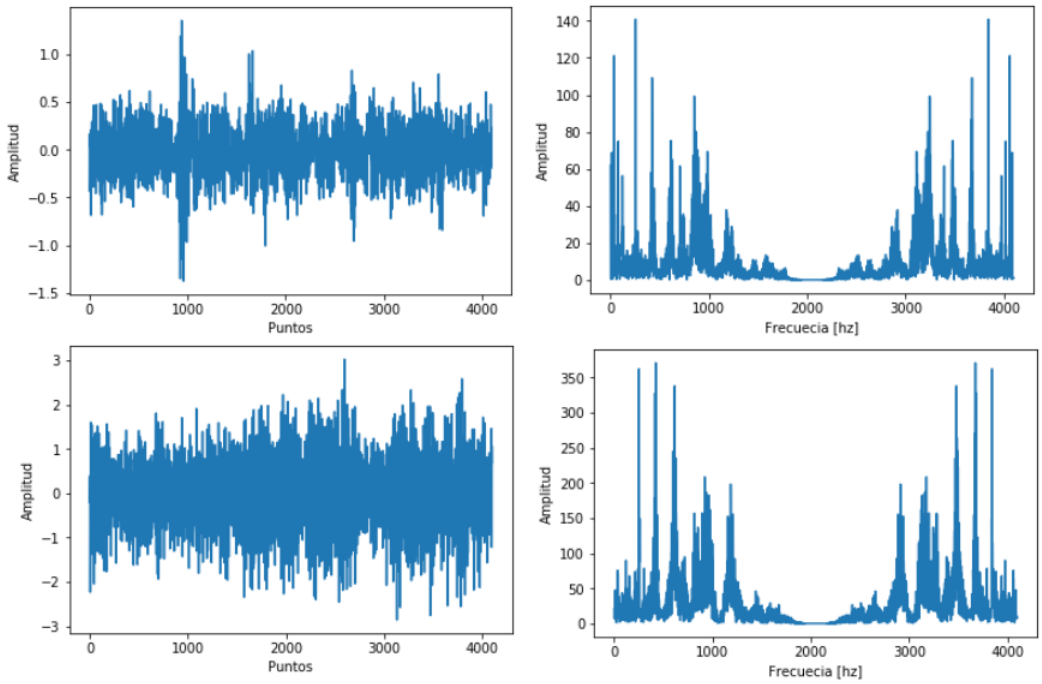


Figura 4.27: Datos de 4,096 puntos y sus transformada de Fourier. Fuente: Elaboración propia.

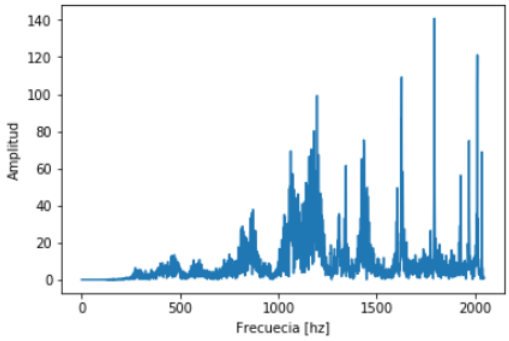


Figura 4.28: Transformada de Fourier de los datos de 2,048 puntos. Fuente: Elaboración propia.

Las Figuras 4.29 y 4.30 se verán más adelante en resultados, debido a que se tienen 2048 dimensiones y no se verán más de 3 dimensiones al mismo tiempo y menos de manera interactiva, por lo que se visualiza una 'matriz' donde se van cruzando ejes/dimensiones de manera respectiva. Se puede ver que las diagonales son el resultado de graficar una dimensión consigo mismo. Finalmente se puede decir que es una matriz casi-simétrica, pues graficar una dimensión 1 con la dimensión 2 es lo mismo que graficar la dimensión 2 con la dimensión 1 pero rotada.

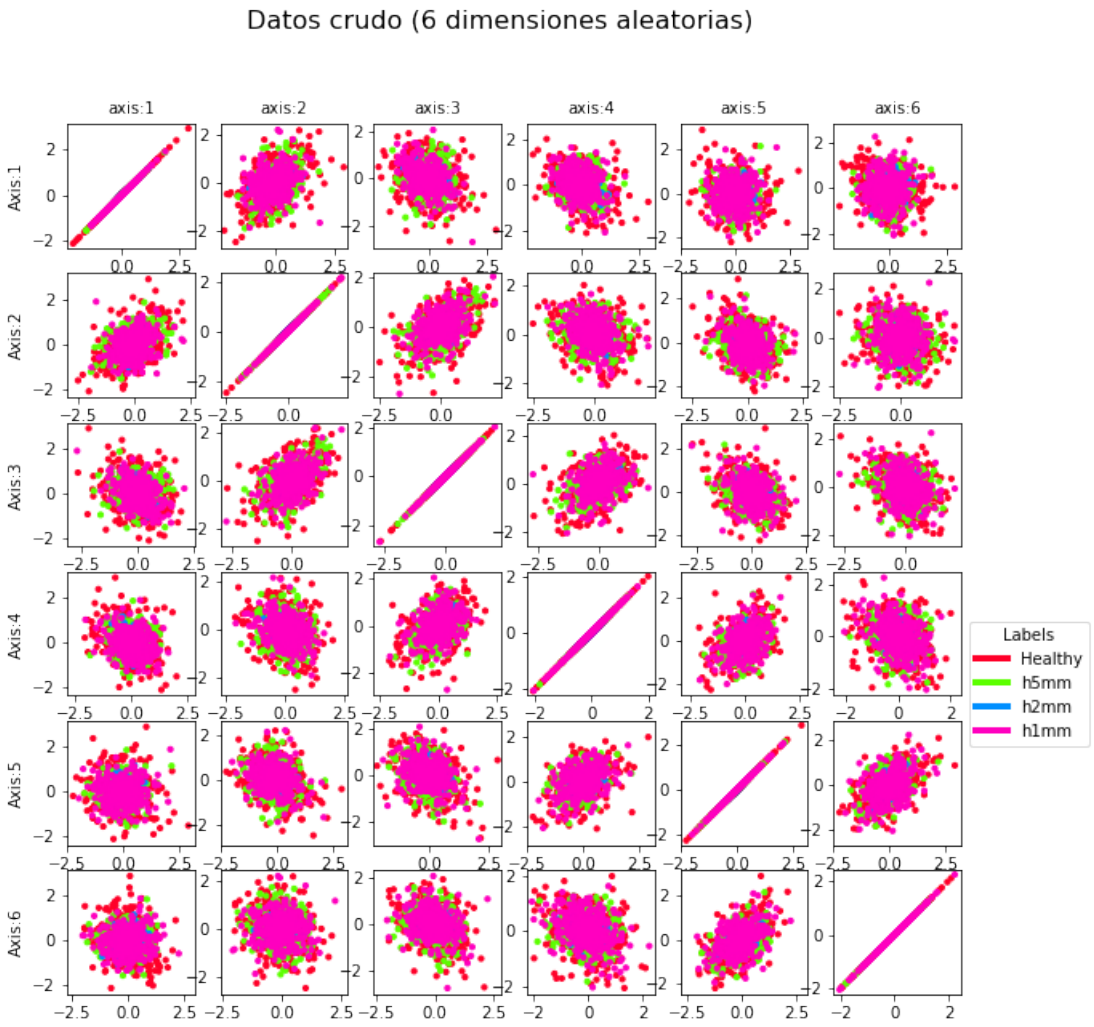


Figura 4.29: Visualización de datos sin procesar. Fuente: Elaboración propia.

4.4. Entrenando para detección de anomalías

Para detección de anomalías, se tiene que realizar un fitting con los datos de entrenamiento, luego los datos de test se utilizan para la detección de anomalías, es decir, los datos de test son los que se utilizan para comparar con un nuevo dato. En estricto rigor, ningún dato de anomalía que se conoce debe ingresarse al entrenamiento de la red.

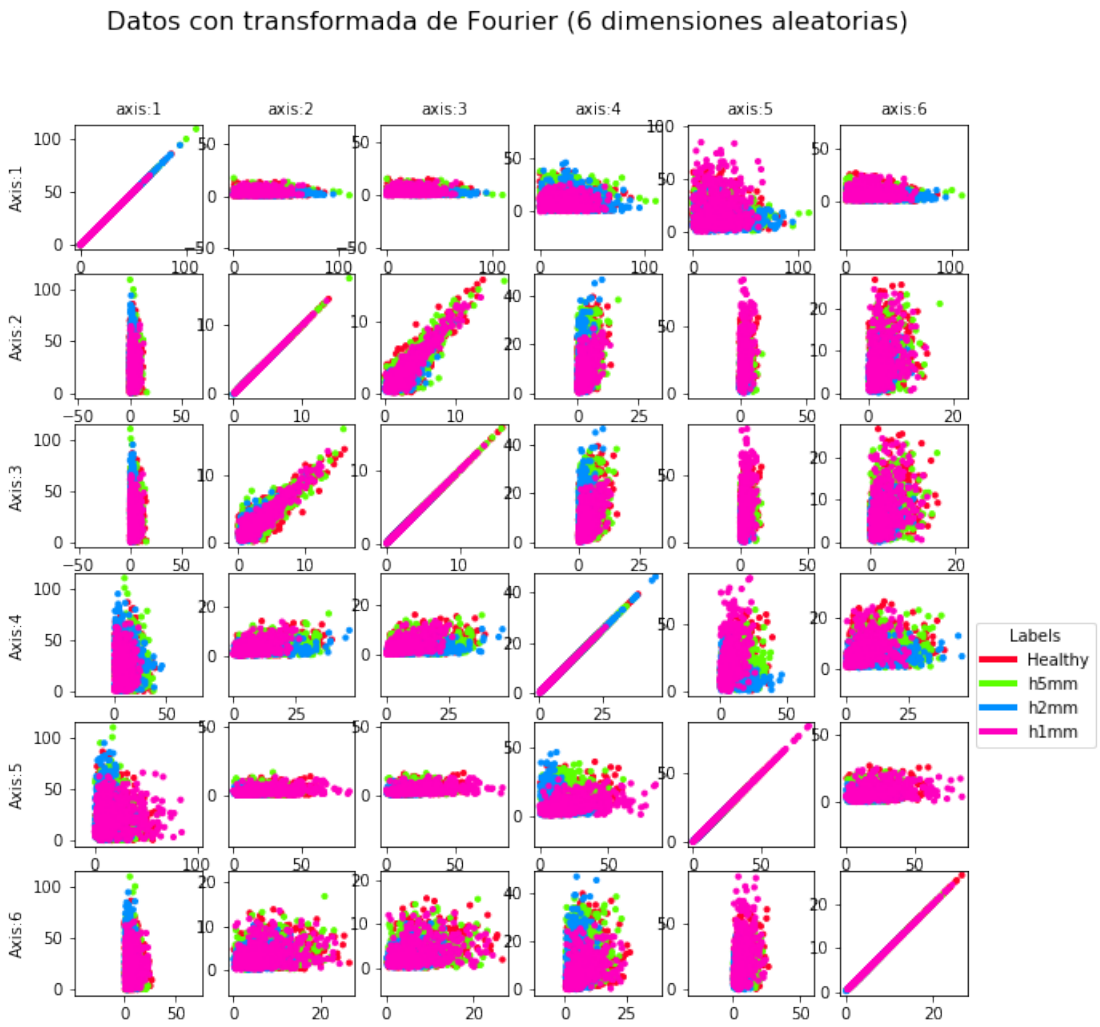


Figura 4.30: Visualización de datos con trasformada de Fourier. Fuente: Elaboración propia.

Capítulo 5

Resultados

En esta sección se mostrarán los resultados de RRCF y RRCF-Rot con los datos experimentales. Se corrieron simulaciones de los siguientes métodos de reducción de dimensión, por lo ya mencionado en la revisión bibliográfica, con la transformada de Fourier:

- PCA.
- AutoEncoder.

Y además se corrieron sin reducción de dimensión para poder ver como afectan los métodos de reducción de dimensión en la precisión de encontrar un punto anómalo. Como se explica en la Sección 2.4.

Para cada prueba se realizaron 3 corridas y se promediaron sus resultados. Primero se mostrará el gráfico de Puntos v/s Rango de CoDISP, luego se resumirán los resultados en una tabla.

5.1. Resultados

En primer lugar se utilizará RRCF y RRCF-Rot sin reducir la dimensión, para poder comparar los cambios en tiempo y precisión con los métodos planteados en la misma tesis.

5.1.1. Sin reducir dimensión

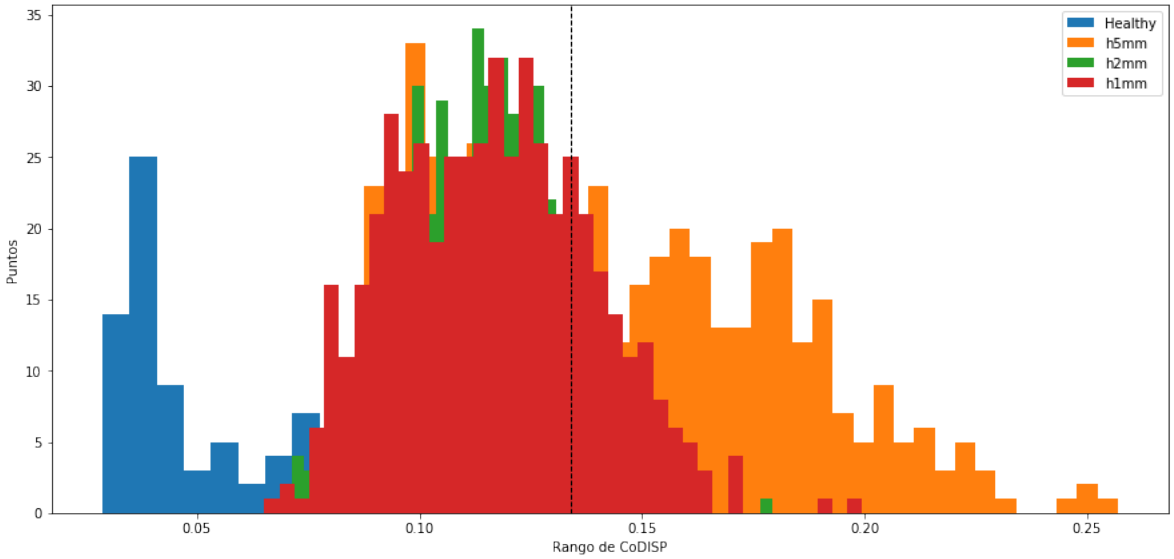


Figura 5.1: Detección de anomalía en RRCF sin reducción de dimensión

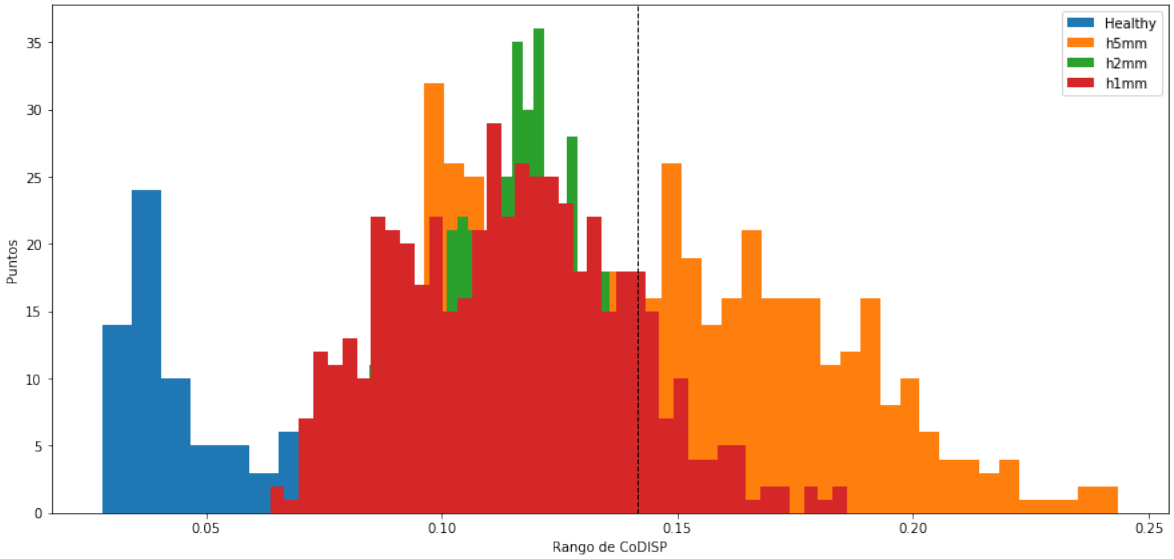


Figura 5.2: Detección de anomalía en RRCF sin reducción de dimensión. Fuente: Elaboración propia.

5.1.2. Procesamiento de datos reduciendo dimensión con PCA

Para poder seleccionar el mejor fitting de PCA sólo se varía el número de componentes finales que se desean. Se comenzó por 2 y se iba sumando para ver el modo en que variaba la detección de anomalías. Reduciendo la dimensión a 2 se obtienen mejores resultados, por lo mismo se puede graficar directamente como se muestra en la Figura 5.4, en su lado izquierdo se muestra solamente el data test y en el lado derecho se muestra el data test con los datos anómalos que debe encontrar.

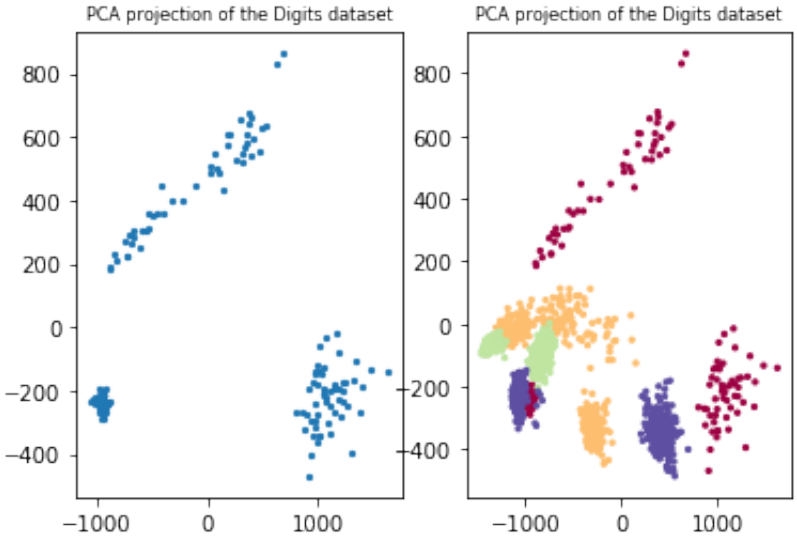


Figura 5.3: Reduciendo dimensión con PCA. Fuente: Elaboración propia.

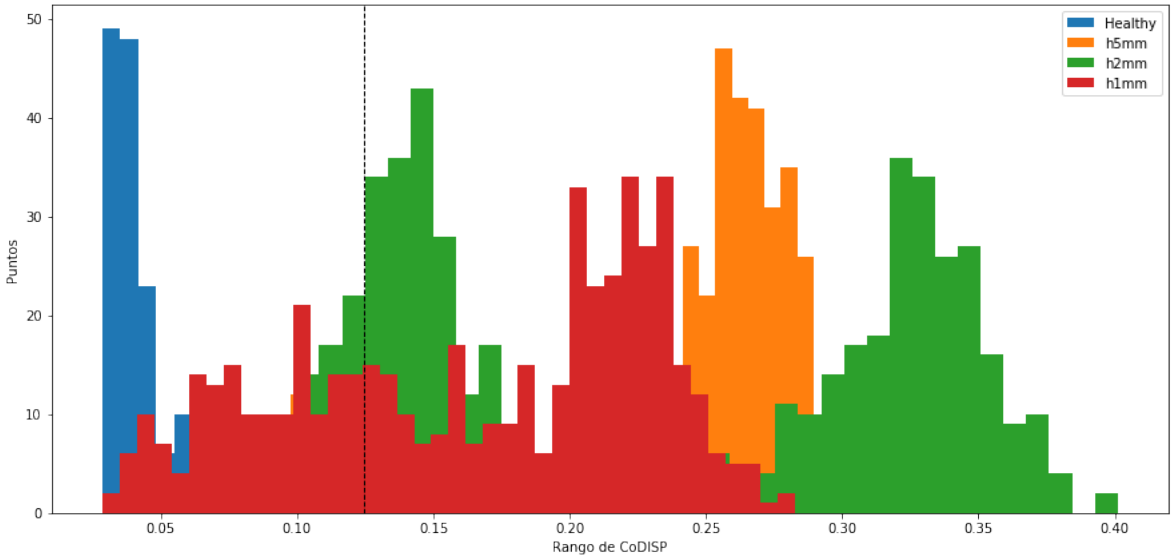


Figura 5.4: Detección de anomalía en RRCF con PCA. Fuente: Elaboración propia.

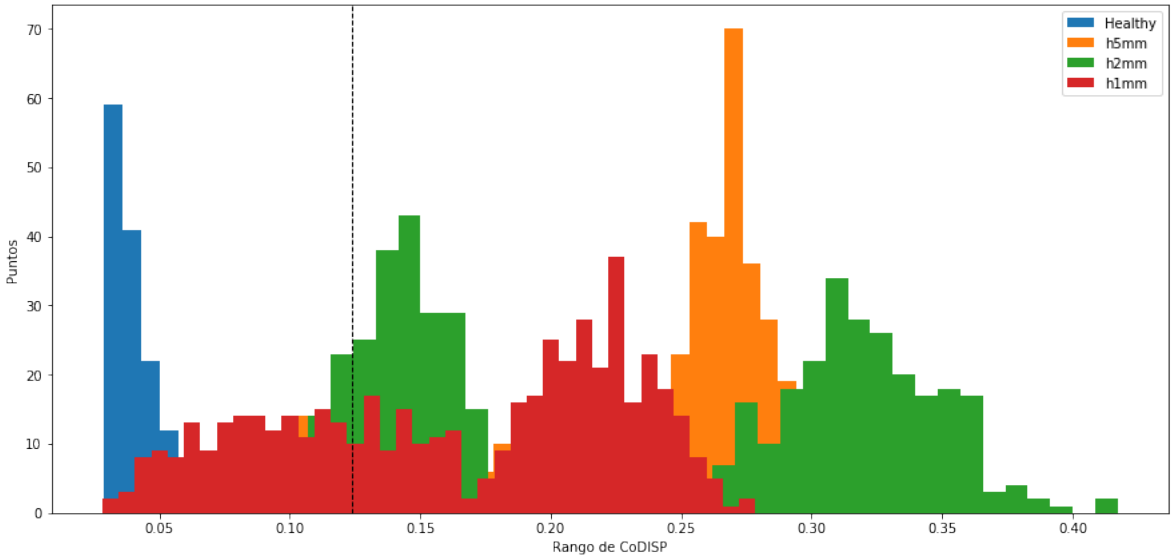


Figura 5.5: Detección de anomalía en RRCF-Rot con PCA. Fuente: Elaboración propia.

5.1.3. Procesamiento de datos en AutoEncoder

La mejor red neuronal fue una con $Z_{latent} = 10$ y una capa de Encoder/Decoder de 100 neuronas. Se entreno por 300 Épocas:
En la Figura 5.6 se puede ver como AutoEncoders puede reconstruir la transformada de Fourier. Tras varias iteraciones se observa que para poder tener un mejor resultado en la detección de anomalías la señal reconstruida se debe parecer lo más posible a la señal de entrada, y también hay que evitar el sobre-entrenamiento. Por lo general no es capaz de conservar numéricamente los peaks más altos de amplitud que en algunos casos puede llegar a 400 de amplitud, mientras en la reconstrucción llega a 120.

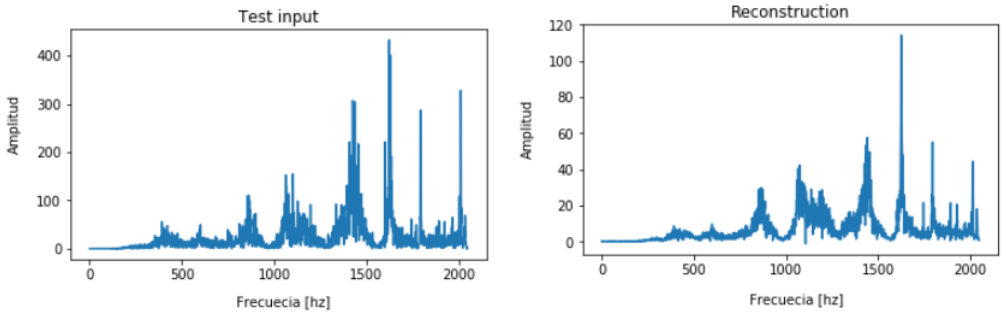


Figura 5.6: Transformadas de Fourier reconstruidas por AutoEncoder. Fuente: Elaboración propia.

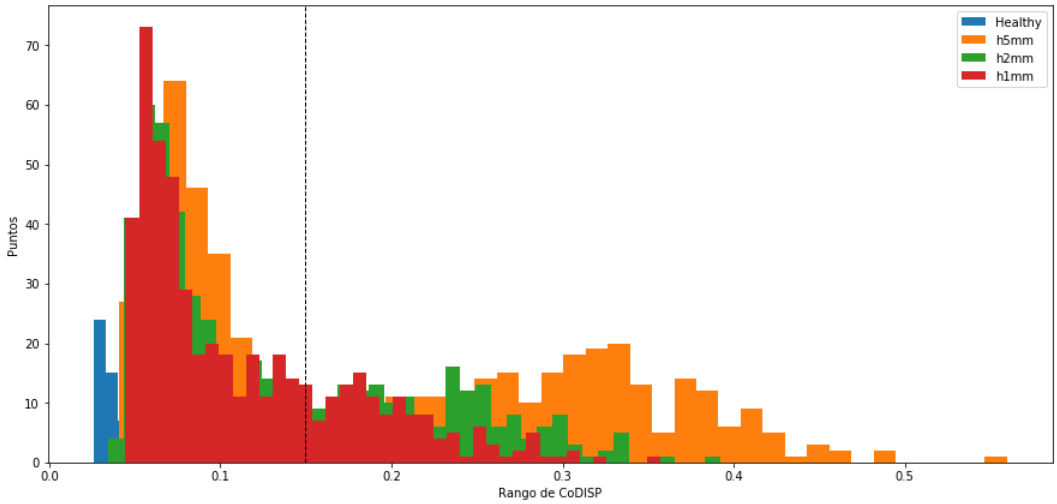


Figura 5.7: Detección de anomalía en RRCF con AutoEncoder. Fuente: Elaboración propia.

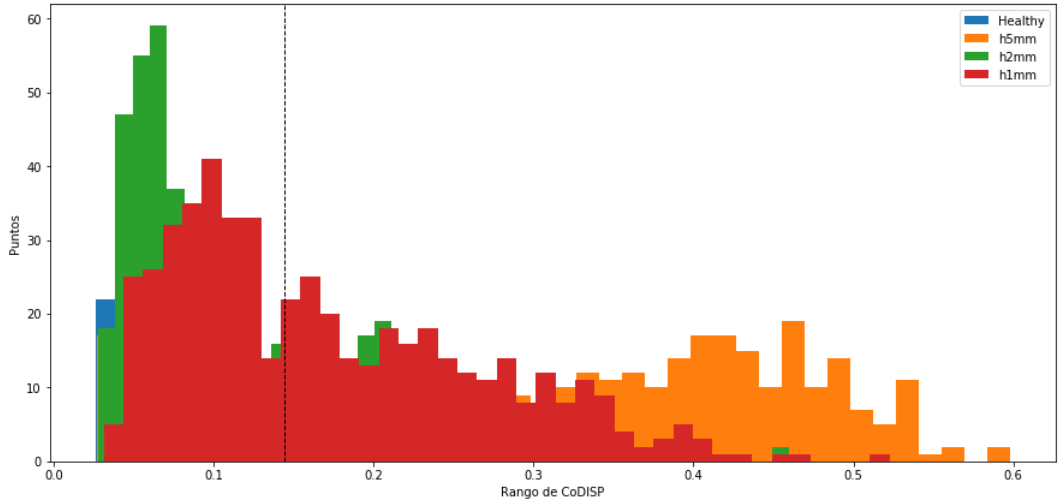


Figura 5.8: Detección de anomalía en RRCF-Rot con AutoEncoder. Fuente: Elaboración propia.

Y esos fueron los gráficos de los métodos de reducción de dimensión utilizados.

5.2. Resumen de resultados

En la Tabla 5.1 se mostrarán los resultados de precisión obtenidos.

Tabla 5.1: Resultados Finales.

Modelo		Ajuste	h5mm		h2mm		h1mm	
		Tiempo[s]	Precisión	Tiempo[s]	Precisión	Tiempo[s]	Precisión	Tiempo [s]
Sin reducción de dimensión	RRCF	11.7	49.5 %	3,516.8	8.65 %	3,755.6	15.5 %	3,675.0
	RRCF-Rot	11.9	48.9 %	3,548.2	6.0 %	3,407.4	12.9 %	3,699.7
Reducción con PCA	RRCF	0.74	85.12 %	61.96	78.64 %	69.90	65.75 %	71.12
	RRCF-Rot	0.75	92.3 %	64.31	93.73 %	67.52	75.8 %	71.15
Reducción con AutoEncoder	RRCF	0.82	60.4 %	123.31	35.8 %	142.01	51.4 %	130.69
	RRCF-Rot	0.84	60.2 %	112.21	34.4 %	133.01	51.8 %	139.04

Conclusión

Se pueden obtener las siguientes conclusiones:

1. Se utilizó RRCF para los datos sin transformada de Fourier, dando resultados nulos, por lo que el uso de la transformada de Fourier aumenta enormemente la precisión de RRCF pues se está usando correctamente el algoritmo según la revisión bibliografía [2].
2. Al agregar un método de reducción de dimensión, mejora la precisión de encontrar una anomalía y se reduce su tiempo de cómputo, esto último se reafirma en la revisión bibliografía [2].
3. La implementación de rotación para cada RRCT, no entrega una mejora en el caso sin reducción de dimensiones, ni con autoencoder. No obstante con el método de reducción con PCA, se puede ver una gran mejora en la precisión. En la Figura 5.2 es posible observar que los datos sin daño se agrupan en tres conjuntos de datos, donde al realizar cada corte del árbol, este logra rotar cada conjunto de datos a su disposición conveniente y realizando una buena detección de anomalías.
4. Mediante la reducción de dimensión con PCA, se obtienen mejores resultados para encontrar datos anómalos para el caso experimental. Luego se probó el Kernel-PCA, pero no se obtuvieron mejores resultados.
5. Al reducir la dimensión en PCA, se llegan a revisar 511 puntos en 71 segundos, es decir, se demora 0.15 segundos aproximadamente para definir si un punto es una anomalía o no. Considerando que el punto debe pasar por una reducción de dimensión estos, después de ser entrenados, no se demoran en reducir dimensión de un dato lo que hace factible realizar un estudio en tiempo real con estos sistemas.
6. El AutoEncoder utilizado en la memoria es uno simple, es decir, existe la posibilidad de utilizar Sparse AutoEncoder, CNN AutoEncoder, etc. No obstante, se probó utilizar Variational AutoEncoder, sin tener un resultado positivo.
7. Finalmente, dicho en las conclusiones anteriores, se puede ver que la reducción de dimensión tanto de PCA y AutoEncoder, mejora tanto en eficiencia (Tiempo de cómputo) como en efectividad (mayor precisión en detectar una anomalía). Por lo que al trabajar con señales de vibración y componentes mecánicos se sugiere utilizar PCA como método de reducción de dimensión y RRCF-Rot para detectar anomalías.

Bibliografía

[1] Sudipto Guha, Nina Mishra, Gourav Roy, and Okke Schrijvers. Robust Random Cut Forest Based Anomaly Detection On Streams. *Proceedings of The 33rd International Conference on Machine Learning*, 48, 2016.

[2] Developer Guide. Amazon SageMaker `https://docs.aws.amazon.com/sagemaker/latest/dg/sagemaker-dg.pdf`.

[3] Robert Bond Randall. Vibration-based Condition Monitoring.

[4] G. E. Hinton* and R. R. Salakhutdinov.

[5] Felipe L Gewers, Gustavo R Ferreira, Henrique F De Arruda, N Filipi, Cesar H Comin, Diego R Amancio, and Luciano F Costa. Principal Component Analysis: A Natural Approach to Data Exploration. pages 1–33.

[6] Aurélien Gerón. Hands-On Machine Learning with Scikit-Learn Tensorflow: Concepts, Tools, and Techniques to build intelligent systems. 2018.

[7] Fei Tony Liu and Kai Ming Ting. Isolation Forest.

[8] Markus Goldstein and Seiichi Uchida. A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data. pages 1–31, 2016.

[9] NCSS Statistical Software. Multidimensional Scaling. pages 1–17.

[10] David M. Chan, Roshan Rao, Forrest Huang, and John F. Canny. T-SNE-CUDA: GPU-Accelerated T-SNE and its Applications to Modern Data. 2018.

[11] Leland Mcinnes, John Healy, and James Melville. UMAP : Uniform Manifold Approximation and Projection for Dimension Reduction. 2018.

[12] Barras planas - A270ES, `http://www.sack.cl/component/virtuemart/barras-tubos-ca~nerías-y-perfiles/16?Itemid=1`. 2018.

[13] Sharon Myers Walpole, Raymond Myers. Probabilidades y Estadística para Ingeniería y Ciencias. Number 816. Pearson, 9. ed edition, 2012.

[14] Sinopiezo - Generador de señal Sinocera de YE1311 http://www.sinopiezo.com/instruments/vibration-excitation-and-calibration-series/sweeping-frequency-signal-generators.html. 2018.

[15] Tektronix - Osciloscopio TDS 210/TDS 220 https://www.tek.com/document/fact-sheet/tds-210-tds-220-fact-sheet. 2018.

[16] globalsensortech -Amplificador de señal Marca SINOCERO modelo YE871A http://www.globalsensortech.com/media/YE5871A.pdf. 2018.

[17] Sinopiezo - Vibrador Sinocera JZK-10 http://www.sinopiezo.com/instruments/vibration-excitation-and-calibration-series/jzk-series-modal-vibration-testing-shakers.html. 2018.

[18] HVT12 Universal Vibration Apparatus http://www.p-a-hilton.co.uk/products/HVT12g-Free-and-Forced-Vibrations-. 2018.

[19] PCB: Sensor model 333B32 http://www.pcb.com/Products/model/333B32. 2018.

Capítulo 6

Apéndice

6.1. Apéndice A:Esperanza Matemática

(Probabilidad y Estádistica para ingeniería y ciencias P112-P115) Se define que la media, o el valor esperado de cualquier variable aleatoria discreta , se puede obtener multiplicando cada uno de los valores $x_1, x_2, ..., x_n$ de la variable aleatoria X por su probabilidad correspondiente $f(x_1), f(x_2), ..., f(x_n)$ y sumando sus productos. En el caso de que sea una variable aleatoria continua la definición de un valor esperado se utiliza la integral.

Definición 6.1 (ver [13]) *Sea* X *una variable aleatoria con distribución de probabilidad* $f(x)$. *La media o valor esperado de* X *es*

$$\mu = \mathbb{E}(\mathrm{X}) = \sum_x xf(x)$$

si X *es discreta, y*

$$\mu = \mathbb{E}(\mathrm{X}) = \int_{-\infty}^{\infty} xf(x)\ \mathrm{d}x$$

si X *es continua.*

Ahora si consideramos una variable aleatoria $g(\mathrm{X})$, es decir cada valor de $g(\mathrm{X})$ depende de X se puede llegar al siguiente teorema:

Teorema 6.2 (ver [13]) *Sea* X *una variable aleatoria con distribución de probabilidad* $f(x)$. *La media o valor esperado de* $g(\mathrm{X})$ *es*

$$\mu_{g(\mathrm{X})} = \mathbb{E}[g(x)] = \sum_x g(\mathrm{X})f(x)$$

si X *es discreta, y*

$$\mu_{g(\mathrm{X})} = \mathbb{E}[g(\mathrm{X})] = \int_{-\infty}^{\infty} g(x)f(x)\ \mathrm{d}x$$

si X *es continua.*

6.2. Distancia de Manhattan

Definición 6.3 *Sean puntos* $P = (x_1, y_1) \in \mathbb{R}$*;* $Q = (x_2, y_2) \in \mathbb{R}$ *se define como una Distancia Manhattan como una función* $\mathbb{R}x\mathbb{R}$ *en* $\mathbb{R}$ *de la siguiente manera.*

$$\mathrm{d_e}(P, Q) = |x_1 - x_2| + |y_1 - y_2| \tag{6.1}$$

Definición 6.4 *Sean puntos* $P = (x_1, y_1) \in \mathbb{R}$*;* $Q = (x_2, y_2) \in \mathbb{R}$ *se define como una Distancia Euclidiana como una función* $\mathbb{R}x\mathbb{R}$ *en* $\mathbb{R}$ *de la siguiente manera.*

$$\mathrm{d_e}(P, Q) = \sqrt{(|x_1 - x_2|)^2 + (|y_1 - y_2|)^2} \tag{6.2}$$

Desigualdad de distancia Manhattan y distancia Euclidiana. Sean puntos $P = (x_1, y_1) \in \mathbb{R}$; $Q = (x_2, y_2) \in \mathbb{R}$ teniendo una distancia euclidiana de P a Q, le sumamos el termino $\sqrt{2 \cdot |x_1 - x_2| \cdot |y_1 - y_2|}$ es fácil ver que es siempre mayor o igual a 0 dependiendo de los terminos de x_1, x_2, y_1 y y_2.

$$\begin{aligned}
\mathrm{d_e}(P, Q) =& \sqrt{(|x_1 - x_2|)^2 + (|y_1 - y_2|)^2} \\
\leq& \sqrt{(|x_1 - x_2|)^2 + (|y_1 - y_2|)^2} + \sqrt{2 \cdot |x_1 - x_2| \cdot |y_1 - y_2|} \\
\leq& \sqrt{(|x_1 - x_2|)^2 + 2 \cdot |x_1 - x_2| \cdot |y_1 - y_2| + (|y_1 - y_2|)^2} \\
\leq& \sqrt{(|x_1 - x_2| + |y_1 - y_2|)^2} \\
\leq& |x_1 - x_2| + |y_1 - y_2| \\
\leq& \mathrm{d}_m(P, Q)
\end{aligned} \tag{6.3}$$

□

6.3. Equipos, instrumentos y materiales utilizados

Para la realización del experimento se utilizaron los siguientes equipos:

- Generador de señal SINOCERA modelo YE1311.
- Osciloscopio Tektronix modelo TDS 210.
- Amplificador de señal SINOCERA modelo YES871A.
- Vibrador SINOCERA modelo JZK-10.

6.3.1. Generador de señal SINOCERA modelo YE1311

Generador de señal de YE1311[14] es un nuevo tipo de generador de señal de barrido que puede producir exactas ondas sinusoidales, ondas cuadradas, ondas triangulares y ruido

blanco. Como una fuente de señal de fines generales se utiliza ampliamente en vibro-acústicos e ingeniería electrónica como prueba modal de vibración, medición de impedancia mecánica, micrófono, altavoz y auriculares calidad análisis, fase de medición de respuesta etc.

Principales especificaciones:

- Salida de onda: sinusoidal, cuadrado, triángulo y ruido blanco.
- Rango de frecuencia: 20Hz - 20 kHz con configuración ajustable de ondas sinusoidal, cuadrada, triángulo, con intervalo fijo para ruido blanco.
- Medidor de frecuencia: Actualización tasa de 20Hz a 20kHz.
- Gama de voltaje de salida 0 a $20V_{rms}$.
- Dimensiones: 360mm(H)x 207mm(W)x 120mm(D).
- Medidor de voltaje rango de frecuencia de operación: 10Hz a 20kHz.
- Peso: 4.0 kg aprox.
- Energía suministro de 220V $\pm$ 10 % 50 Hz.

6.3.2. Osciloscopio Tektronix modelo TDS 210

Es un osciloscopio[15] con dos canales de entradas. Sus principales especificaciones son:

- Ancho de banda: 60 MHz.
- Frecuencia de muestreo: 1 GS/sec.
- Largo de grabación: 2500 puntos por canal.
- Vertical: 2 mV to 5 V per division.
- Horizontal: 5 ns a 5 s.
- Dimensiones: 151mm(H)x 305mm(W)x 110mm(D).
- Peso: 2.9 kg

6.3.3. Amplificador de señal SINOCERA modelo YES871A

El amplificador de señal SINOCERA modelo YES871A[16] tiene una particularidad de poder trabajar con vibrador SINOCERA de modelo JZK-10. Sus principales características son:

- Indica l el voltaje y la corriente de salida.
- Indica la señal de salida.
- Corriente de salida se ajusta de 2A a 10A.
- Salida constante de voltaje y corriente.

6.3.4. Vibrador SINOCERA modelo JZK-10

Los vibradores de la serie JZK[17] son transductores eléctrico-mecánicos que convierten energía eléctrica en energía mecánica para proporcionar una vibración controlada de la fuerza que se aplica a una estructura o componente de la prueba. Se utiliza una serie YE587x o similar tipo amplificadores y otros equipos para medir la respuesta de la vibración de una estructura de amplia gama en defensa e ingeniería civil. Principales especificaciones del JZK-10:

- Max. viajes:±5 (mm).
- Max. Aceleración: 280 (ms-2).
- Rango de frecuencia: 10 a 4000 (Hz).
- Eficaz movimiento: 0.35 (kilogramo).
- Amplificador de Energía emparejada: YE5871 y YE5872.
- Dimensiones: ϕ 158 150(mm).

Para la realización del experimento se utilizaron los siguientes instrumentos:

- Marco de vibraciones de P.A.Hilton HVT12f (HVT12 Universal Vibration Apparatus).
- Sensor PCB modelo 333B32.
- Barras de acero.

6.3.5. Marco de vibraciones de P.A.Hilton HVT12f

El marco de vibraciones de P.A.Hilton HVT12f[18] es un marco de acero en el cual se pueden instalar distintos componentes, como resortes, pivotes, cubetas y motores. Se utiliza para hacer estudios de vibración.

6.3.6. Sensor PCB modelo 333B32

Sensor PCB modelo 333B32[19] utilizado para medir la aceleración. Sus principales características son:

- Sensibilidad: ($\pm$10 %) 10.2 mV/(m/s^2)
- Rango de medición: $\pm$ 490 m/s^2 pk
- Resolución: 0.0015 m/s^2 rms
- Rango de frecuencia: ($\pm$ 5 %) 0.5 to 3000 Hz
- Elemento del sensor: Cerámica.
- Peso: 4 gramos