



INSTITUTO TECNOLÓGICO SUPERIOR DE MISANTLA

---

MACHINE LEARNING PARA CLASIFICACIÓN DE  
NEUROPATÍA DIABÉTICA A TRAVÉS DE BIOMARCADORES  
DE LA MARCHA

TESIS

PARA OBTENER EL GRADO DE:  
**Maestra en Sistemas Computacionales**

Presenta:

**I.S.C Fabiola Monrraga Bernardino**

Director:

Dr. Eddy Sánchez de la Cruz

Co-Director:

Dr. Iván Vladimir Meza Ruiz

Misantla, Veracruz Agosto de 2018

# Autorización de impresión

## Dedicatoria

*Mis padres Victoria y Ermeregildo, gracias a su apoyo incondicional por siempre apoyarme en mis proyectos en confiar en mí, por los consejos que me brindaron, gracias a la formación que me dieron y a la educación que me brindaron estoy concluyendo un logro muy importante para mí. A mis hermanas Leticia y Alicia por estar siempre a mi lado, por darme la alegría de ser querida por ustedes, gracias por su amor incondicional que me han brindaron y consejos que me han dado. A mi novio Roldolfo gracias por estar conmigo que a pesar de la distancia nuestro amor sigue adelante y siempre me has apoyado en todo momento.*

## **Agradecimientos**

Gracias Dios por la salud y bienestar que me has brindado, gracias por ser afortunada de tener una gran familia que siempre ha confiado en mi y me ha apoyado.

Agradezco a los Doctores Eddy e Iván por haberme brindado la oportunidad de recurrir a su capacidad y conocimiento, además por haberme tenido toda la paciencia para guiarme durante el desarrollo de la tesis.

Doctor Weber gracias por brindarme el apoyo en el tiempo de estadía en el Servicio médico de la Universidad Juárez Autónoma de Tabasco y así mismo el personal de la clínica que fueron muy amables conmigo.

IIMAS (Instituto de Investigación de Matemáticas Aplicadas) por brindarme el apoyo durante el tiempo de estancia que estuve trabajando con ustedes.

CONACYT (Consejo Nacional de Ciencia y Tecnología.) por brindarme el apoyo de poder realizar y concluir la maestría.

## Resumen

La neuropatía diabética es un padecimiento que afecta a la población adulta con diabetes tipo I y II (Ramírez-López, Giles, y Avilés, 2017). Existen diferentes síntomas que reflejan esta patología, como dolor de las articulaciones, hormigueo en los pies, dolor al estirar las piernas, dolor al caminar y calambres (Molina, Rodríguez, y cols., 2012). En la presente tesis se propone una red de sensores como apoyo al especialista, para el diagnóstico de neuropatía diabética. Se obtienen biomarcadores de la marcha de la espalda baja, rodilla derecha y rodilla izquierda, tobillo derecho y tobillo izquierdo, dicha información se obtuvo de sujetos de estudio de la Unidad de Servicios Médicos de la Universidad Juárez Autónoma de Tabasco, bajo la supervisión del médico Roberto Weber Burque Cruz, quien determinó si el paciente se encontraba sano o sufría de neuropatía diabética. La población fue de diez pacientes con padecimiento y cinco controles sanos. El análisis de la información se realizó como sigue: primero se hizo el preprocesamiento de los datos para la eliminación de ruido, inconsistencia y redundancia, luego, se realizó el análisis tres algoritmos de Machine learning para obtener el mejor en clasificación de datos. Los resultados mostraron que el algoritmo AdaBoost obtuvo una precisión del 87%.

## Abstract

Diabetic neuropathy is a condition that affects the adult population with type I and II diabetes *et al* (Ramírez-López y cols., 2017). There are different symptoms that reflect this pathology, such as joint pain, tingling in the feet, pain when stretching the legs, pain when walking and cramps *et al* (Molina y cols., 2012). In the present thesis a network of sensors is proposed as support to the specialist, for the diagnosis of diabetic neuropathy. Biomarkers are obtained from the gait of the lower back, the right leg, knee and ankle and left leg, knee and ankle. This information was obtained from study subjects of the Medical Services Unit of the Juárez Autónoma de Tabasco University, under the supervision of the Dr. Roberto Weber Burque Cruz, who determined whether the patient was healthy or suffering from diabetic neuropathy. The population was ten patients with disease and five healthy controls. The analysis of the information was made as follows: first, the data was pre-processed for noise elimination, inconsistency and redundancy, then the analysis was carried out using three Machine Learning algorithms to obtain the best data classification. The results showed that the algorithm AdaBoost obtained a precision of 87%.

# Índice general

<b>Autorización de impresión</b>	<b>II</b>
<b>Dedicatoria</b>	<b>III</b>
<b>Agradecimientos</b>	<b>IV</b>
<b>Resumen</b>	<b>V</b>
<b>Abstract</b>	<b>VI</b>
<b>Índice de figuras</b>	<b>IX</b>
<b>Índice de tablas</b>	<b>X</b>
<b>Índice de algoritmos</b>	<b>XI</b>
<b>1. Generalidades</b>	<b>12</b>
1.1. Introducción . . . . .	13
1.2. Planteamiento del problema . . . . .	13
1.2.1. Justificación . . . . .	14
1.3. Objetivos . . . . .	14
1.3.1. Objetivo general . . . . .	14
1.3.2. Objetivos específicos . . . . .	14
1.4. Hipótesis . . . . .	14
1.5. Propuesta de solución . . . . .	15
1.6. Alcances y limitaciones . . . . .	16
1.6.1. Alcances . . . . .	16
1.6.2. Limitaciones . . . . .	16
1.7. Estructura de la tesis . . . . .	16
<b>2. Marco teórico</b>	<b>17</b>
2.1. Definiciones y conceptos . . . . .	18
2.1.1. Diabetes mellitus . . . . .	18
2.1.2. Neuropatía diabética . . . . .	18

2.1.3.	Machine learning . . . . .	18
2.1.4.	Minería de datos . . . . .	19
2.1.5.	Cross-validation . . . . .	19
2.1.6.	Matriz de confusión . . . . .	20
2.1.7.	Medidas de evaluación de clasificadores . . . . .	20
2.2.	Algoritmos de clasificación . . . . .	22
2.2.1.	Algoritmo Random Forest . . . . .	22
2.2.2.	Algoritmo Support Vector Machine . . . . .	22
2.2.3.	Algoritmo Adaboost . . . . .	23
2.2.4.	Algoritmo LogitBoost . . . . .	24
2.3.	Herramientas computacionales . . . . .	25
2.3.1.	Hardware . . . . .	25
2.3.2.	Software . . . . .	26
<b>3.</b>	<b>Estado del arte</b>	<b>28</b>
3.1.	Trabajos relacionados con la Red de sensores . . . . .	29
3.2.	Trabajos relacionados con la clase de ND . . . . .	31
<b>4.</b>	<b>Metodología</b>	<b>34</b>
<b>5.</b>	<b>Experimentos y resultados</b>	<b>36</b>
5.1.	Experimento I . . . . .	37
5.1.1.	Construcción del Sensor rodilla-tobillo . . . . .	38
5.1.2.	Recolección de datos . . . . .	39
5.1.3.	Análisis de datos . . . . .	40
5.1.4.	Resultados . . . . .	40
5.2.	Experimento II . . . . .	41
5.2.1.	Construcción de la red de sensores . . . . .	41
5.2.2.	Construcción del DataSet . . . . .	42
5.2.3.	Estructura del dataset . . . . .	44
5.2.4.	Preprocesamiento de datos . . . . .	44
5.2.5.	Conjunto de entrenamiento . . . . .	46
5.2.6.	Resultados . . . . .	46
<b>6.</b>	<b>Análisis de resultados</b>	<b>49</b>
<b>7.</b>	<b>Conclusiones y Trabajos Futuros</b>	<b>65</b>
7.1.	Conclusiones . . . . .	66
7.2.	Trabajos Futuros . . . . .	66
	<b>Referencias</b>	<b>67</b>
	<b>Anexos</b>	<b>69</b>

# Índice de figuras

1.1. Proceso de clasificación de datos . . . . .	15
2.1. Matriz de confusión . . . . .	21
5.1. Figura proceso de clasificación. . . . .	37
5.2. Topología del sensor rodilla/ tobillo. . . . .	38
5.3. Colocación de la red sensores. . . . .	39
5.4. Algoritmos Metaclasificadores . . . . .	41
5.5. Topología de la red sensores. . . . .	42
5.6. Prueba previas de sensores. . . . .	43
5.7. Servicios médicos UJAT . . . . .	43
5.8. Ubicación de la red sensores. . . . .	44
5.9. Conjunto de datos en formato .csv . . . . .	45
5.10. Datos de Entrenamiento y prueba . . . . .	47
6.1. Paciente 01 . . . . .	50
6.2. Paciente 02 . . . . .	51
6.3. Paciente 03 . . . . .	52
6.4. Paciente 04 . . . . .	53
6.5. Paciente 05 . . . . .	54
6.6. Paciente 06 . . . . .	55
6.7. Paciente 07 . . . . .	56
6.8. Paciente 08 . . . . .	57
6.9. Paciente 09 . . . . .	58
6.10. Paciente 10 . . . . .	59
6.11. Paciente 11 . . . . .	60
6.12. Paciente 12 . . . . .	61
6.13. Paciente 13 . . . . .	62
6.14. Paciente 14 . . . . .	63
6.15. Paciente 15 . . . . .	64

# Índice de tablas

5.1. Materiales y costo del sensor rodilla-tobillo . . . . .	38
5.2. Información de sujetos de pruebas . . . . .	40
5.3. Matriz de confusión . . . . .	41
5.4. Materiales y costos . . . . .	42
5.5. Información de pacientes. . . . .	45
5.6. Matriz de confusión por paciente . . . . .	47
5.7. Matriz de confusión por segmento . . . . .	47
5.8. Algoritmo AdaBoost. . . . .	48
5.9. Algoritmo RandomForest . . . . .	48
5.10. Algoritmo SVM . . . . .	48

# Índice de algoritmos

7.1. Código de obtención de datos de la red de sensores . . . . .	69
7.2. Código en PYTHON para la clasificación de algoritmos . . . . .	70

# Capítulo 1

## Generalidades

## 1.1. Introducción

La presente tesis aborda el estudio de la clasificación de biomarcadores de la marcha, los cuales se obtienen mediante una red de sensores. La neuropatía diabética (ND) se debe al daño a los nervios con padecimiento a la diabetes y algunos de los síntomas que producen entumecimiento, hormigueo, dolor debilidad que generalmente comienza en los pies y progresa en las piernas, su prevalencia se incrementa con los años en padecer diabetes mellitus tipo II. Entre las características principales de la neuropatía diabética se encuentran: la excesiva formación de sorbitol en las células nerviosas, la disminución de mionisal, el incremento de la glicación de los nervios periféricos y la insuficiencia muscular, que lleva al daño microvascular e hipoxia.

Los objetivos principales del presente estudio fueron: construir una red de sensores de bajo costo para la captura de biomarcadores de la marcha y el diseño de un modelo de clasificación de ND utilizando técnicas de aprendizaje automático y minería de datos que proporcione la clasificación correcta, para su posible uso en futuras investigaciones en, por ejemplo, enfermedades neurodegenerativas. Para alcanzar el objetivo de la investigación se llevaron acabo las siguientes tareas de importancia:

- construir una red de sensores que capture biomarcadores de la marcha,
- crear un registro en tiempo real de biomarcadores de la marcha,
- analizar las características obtenidas,
- realizar el pre-procesamiento de datos,
- analizar los diferentes algoritmos de aprendizaje automático y, por ultimo,
- determinar el algoritmo de clasificación óptimo.

## 1.2. Planteamiento del problema

La diabetes mellitus es uno de los problemas más graves de salud pública a nivel mundial (y en México) la cual es la segunda causa de mortalidad del 15.4% de la población<sup>1</sup>. Se calcula las personas entre 20 a 69 años de edad la padecen. En México se reporta que 38% de los individuos afectados tiene alguna forma de ND y su frecuencia depende de la duración de la diabetes.

La ND se refiere a la alteración de los nervios con trastornos en la sensibilidad. Se pueden clasificar en dos tipos de neuropatías: periférica (extremidades) y visceral (órganos o funciones internas). La ND se debe a una lesión de los nervios a consecuencia de la diabetes, puede incluso llegar a afectar al 50% de los pacientes <sup>2</sup>.

---

<sup>1</sup><http://www.inegi.org.mx>

<sup>2</sup><http://www.oment.uanl.mx>

Desde una perspectiva computacional, se puede identificar correctamente la ND, obteniendo biomarcadores de la marcha de sujetos que la padecen y comparándolos con los de sujetos sanos, esto mediante la implementación de algoritmos de clasificación.

### **1.2.1. Justificación**

De los trabajos relacionados en las áreas de computación y medicina para la captura de información durante la marcha del paciente, la mayoría de los dispositivos solo se encuentran enfocados en una parte específica del cuerpo y la información que se obtiene es mínima para realizar un análisis y dar un diagnóstico. Actualmente existen pocas investigaciones sobre el área de neuropatía diabética que involucren obtener información de biomarcadores de la marcha y se capturen en las partes de mayor importancia, es decir, las piernas del paciente: rodilla, tobillo y espalda baja o centro de gravedad. Por lo anterior, desarrollar una red de sensores para obtener biomarcadores de la marcha y realizar el análisis de diferentes algoritmos utilizando aprendizaje automático y minería de datos para una clasificación correcta de neuropatía diabética, es una opción viable.

## **1.3. Objetivos**

### **1.3.1. Objetivo general**

Desarrollar un modelo para la clasificación de la marcha aplicando técnicas de aprendizaje automático en biomarcadores de neuropatía diabética.

### **1.3.2. Objetivos específicos**

1. Diseñar una red de sensores para la captura de biomarcadores de la marcha.
2. Validar la red de sensores.
3. Recolectar biomarcadores de la marcha a pacientes con neuropatía diabética y sujetos sanos.
4. Realizar el pre-procesamiento de la información adquirida.
5. Realizar pruebas con diferentes algoritmos de aprendizaje automático y seleccionar el de mejor desempeño.
6. Analizar y validar los resultados.

## **1.4. Hipótesis**

*Es posible clasificar correctamente sujetos sanos de personas que padecen ND, mediante la implementación de algoritmos de aprendizaje automático sobre biomarcadores de la marcha.*

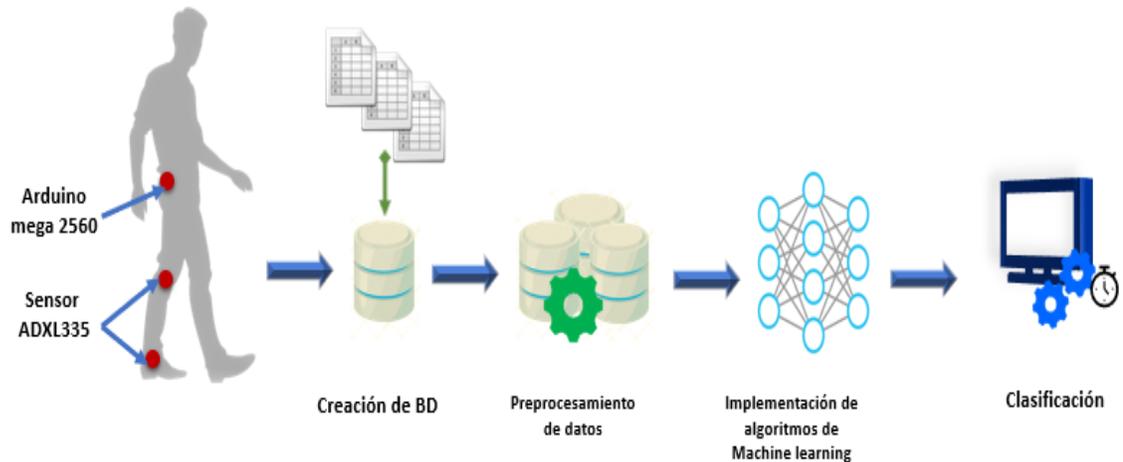


Figura 1.1: Proceso de clasificación de datos

## 1.5. Propuesta de solución

Para dar solución al problema de neuropatía diabética, se propone un diseño de clasificación para el estudio al que pertenece el paciente basados en técnicas de aprendizaje automático que servirá para analizar y clasificar los datos que se muestran en la Figura 1.1.

A continuación se describe cada uno de los puntos que correspondiente a la clasificación de datos.

1. **Data set:** los datos adquiridos a partir de los pacientes con neuropatía diabética y pacientes sanos que asistieron a consultan en la servicios médicos de la Universidad Juárez Autónoma de Tabasco (UJAT).
2. **Preprocesamiento de datos:** selección relevante de datos, eliminando registros duplicados, anomalías, para mejorar la calidad del conjunto de datos y se obtenga mayor y mejor información.
3. **Implementación de algoritmos de Machine learning:** realizar el estudio de los diferentes algoritmos de clasificación y encontrar el óptimo para los datos recolectados.
4. **Clasificación:** consiste en agrupar los datos en clases predefinidas de acuerdo a ciertas características que tienen en común. Para el proceso de clasificación se utiliza el subconjunto de características previamente seleccionadas, y a partir de dichas características se encuentran patrones que determinan la clase en la que pertenecen.

## **1.6. Alcances y limitaciones**

### **1.6.1. Alcances**

Desarrollar una red de sensores de bajo costo y sencilla de manejar.  
Identificar el algoritmo de clasificación adecuado para resolver la problemática.

### **1.6.2. Limitaciones**

Obtener suficientes pacientes para la construcción de un data-set con personas de enfermedades neurodegenerativas.

## **1.7. Estructura de la tesis**

El contenido de la investigación se ha dividido en los siguientes capítulos.

1. El primer capítulo se encuentra el protocolo de la investigación integrado planteamiento de problema, objetivo, justificación, hipótesis, propuesta de solución.
2. El segundo capítulo contiene definiciones, conceptos que se tiene relación al tema el cual esta dividido dos partes, (I) área medica, (II) área tecnológica.
3. El tercer capítulo el estado del arte el cual consta de en dos partes investigaciones, la primera prototipos realizados para la obtención de información de personas con neuropatía diabética y la segunda algoritmos óptimos para la clasificaciones en aprendizaje automático.
4. El cuarto capítulo muestra, los experimentos y las etapas realizadas para la clasificación de los datos, guiados por cada una de las etapas de la metodología de solución.
5. El capítulo cinco contiene el análisis de resultados y las conclusiones que se demuestran con la investigación.

# **Capítulo 2**

## **Marco teórico**

Las áreas de computación y medica han trabajado en conjunto para el desarrollo de sistemas que apoyen en el monitoreo, control, predicción, tratamiento en la mayoría de las enfermedades. En el caso de las enfermedades neurodegenerativas existen diferentes dispositivos en predicción y métodos de diagnostico.

A continuación se explicaran los conceptos mas importantes realizados en este trabajo.

## **2.1. Definiciones y conceptos**

### **2.1.1. Diabetes mellitus**

La diabetes mellitus se define como una enfermedad crónico-degenerativa, ocasionada por diversos factores, y entre sus diferentes tipos se toman en cuenta factores como la predisposición hereditaria, los factores ambientales y de estilos de vida, así como que se caracteriza por hiperglucemia crónica (altos niveles de azúcar en sangre por periodos prolongados) debido a la deficiencia en la producción o acción de la insulina, lo que afecta al metabolismo de los carbohidratos, proteínas y grasas <sup>1</sup>.

### **2.1.2. Neuropatía diabética**

La Neuropatía Diabética (ND), se define como la presencia de síntomas y/o signos de disfunción nerviosa periférica en las personas con diabetes, una vez que han sido eliminadas otras causas, el diagnóstico puede presuponerse cuando dos o más signos anormales en la función del sistema nervioso periférico están presentes. En la literatura las prevalencias reportadas oscilan del 10 al 90%. La ND se ha implicado como directamente causal del 50-70% de las amputaciones no traumáticas. Aproximadamente 30 millones de pacientes en todo el mundo sufren alguna forma de ND, la complicación más frecuente tanto de la diabetes mellitus tipo I como de la tipo II, por lo que el padecimiento se considera un importante problema de salud pública. La ND está presente en un 40 a 50% después de 10 años del comienzo de la enfermedad, tanto en los tipo I como en los tipo II, aunque menos del 50% de estos pacientes presentan síntomas. Su prevalencia aumenta con el tiempo de evolución de la enfermedad y con la edad del paciente, relacionándose su extensión y gravedad con el grado y duración de la hiperglucemia (Ramírez-López y cols., 2017).

### **2.1.3. Machine learning**

El Aprendizaje Automático (del inglés, Machine Learning) tiene como objetivo desarrollar sistemas (programas de computadora) de forma que las computadoras aprendan, o sea, que puedan mejorar su funcionamiento para realizar una tarea de forma automática, basándose en la experiencia (Díaz, 2014). El aprendizaje automático ofrece una alternativa más eficiente para capturar el conocimiento en datos para mejorar gradualmente el rendimiento de los

---

<sup>1</sup><http://www.inegi.org.mx/inegi/default.aspx?c=274>

modelos predictivos y tomar decisiones basadas en datos. El aprendizaje automático no solo es cada vez más importante en la investigación en informática, sino que también desempeña un papel cada vez más importante en nuestra vida cotidiana (Raschka, 2015).

Al diseñar un sistema de aprendizaje se deben tomar en cuenta varios elementos: el tipo de experiencia de entrenamiento, la medida de desempeño, la función objetivo y su representación así como el algoritmo para aproximarla.

En función de la clase de experiencia utilizada, el Aprendizaje Automático puede dividirse en tres tipos principales (Díaz, 2014):

- **Supervisado:** es aprender un modelo a partir de los datos de entrenamiento etiquetados que nos permite hacer predicciones sobre datos ocultos o futuros. Aquí, el término supervisado se refiere a un conjunto de muestras donde ya se conocen las señales de salida deseadas (etiquetas).
- **No supervisado:** consiste en aprender patrones de entradas cuando no hay valores de salida especificados. Las clases se infieren de los datos, creando grupos diferenciados.
- **Refuerzo:** el objetivo es desarrollar un sistema (agente) que mejore su rendimiento en función de las interacciones con el entorno.

#### 2.1.4. Minería de datos

Minería de datos (DM, por sus siglas en inglés Data Mining) es una etapa del descubrimiento del conocimiento en base de datos (KDD, por sus siglas en inglés Knowledge Discovery in Databases) y se encarga de extraer de manera automática información a partir de los datos recopilados (Díaz, 2014). El concepto de Minería de Datos se traduce como extracción de datos, análisis de datos, recopilación de datos. Debido a la gran variedad de tipos de datos y formas de organizar la información, es posible que los datos reales no siempre sean analizados por las herramientas de aprendizaje automático. Para la transformación de datos “en bruto” a los datos, que pueden funcionar de manera eficiente las técnicas de minería de datos, resuelven el problema del preprocesamiento (Vadim, 2018).

#### 2.1.5. Cross-validation

La validación cruzada o cross-validation es una técnica utilizada para evaluar los resultados de un análisis estadístico y garantizar que son independientes de la partición entre datos de entrenamiento y prueba. Consiste en repetir y calcular la media aritmética obtenida de las medidas de evaluación sobre diferentes particiones. Se utiliza en entornos donde el objetivo principal es la predicción y se quiere estimar la precisión de un modelo que se llevará a cabo a la práctica. Es una técnica muy utilizada en proyectos de inteligencia artificial para validar modelos generados (Witten, 2005).

1. **Validación cruzada de K iteraciones** o K-fold cross-validation los datos de muestra se dividen en K subconjuntos. Uno de los subconjuntos se utiliza como datos de prueba

y el resto (K-1) como datos de entrenamiento. El proceso de validación cruzada es repetido durante k iteraciones, con cada uno de los posibles subconjuntos de datos de prueba. Finalmente se realiza la media aritmética de los resultados de cada iteración para obtener un único resultado. Este método es muy preciso puesto que evaluamos a partir de K combinaciones de datos de entrenamiento y de prueba, pero aun así tiene una desventaja, y es que, a diferencia del método de retención, es lento desde el punto de vista computacional. En la práctica, la elección del número de iteraciones depende de la medida del conjunto de datos. Lo más común es utilizar la validación cruzada de 10 iteraciones (10-fold cross-validation)

2. **Validación cruzada aleatoria** el método consiste al dividir aleatoriamente el conjunto de datos de entrenamiento y el conjunto de datos de prueba. Para cada división la función de aproximación se ajusta a partir de los datos de entrenamiento y calcula los valores de salida para el conjunto de datos de prueba. El resultado final se corresponde a la media aritmética de los valores obtenidos para las diferentes divisiones. La ventaja de este método es que la división de datos entrenamiento-prueba no depende del número de iteraciones. Pero, en cambio, con este método hay algunas muestras que quedan sin evaluar y otras que se evalúan más de una vez, es decir, los subconjuntos de prueba y entrenamiento se pueden solapar.
3. **Validación cruzada dejando uno fuera** o Leave-one-out cross-validation (LOOCV) implica separar los datos de forma que para cada iteración tengamos una sola muestra para los datos de prueba y todo el resto conformando los datos de entrenamiento. La evaluación viene dada por el error, y en este tipo de validación cruzada el error es muy bajo, pero en cambio, a nivel computacional es muy costoso, puesto que se tienen que realizar un elevado número de iteraciones, tantas como N muestras tengamos y para cada una analizar los datos tanto de entrenamiento como de prueba.

### 2.1.6. Matriz de confusión

La matriz de confusión permite observar el desempeño de un algoritmo donde cada columna de la matriz representa un número de predicciones de cada clase, mientras que las filas representan las instancias en la clase real como se muestra en la Figura. 2.1.

### 2.1.7. Medidas de evaluación de clasificadores

La efectividad de un clasificador con respecto a un conjunto de datos es usualmente comprobada usando un conjunto de medias de evaluación (también denominadas funcionales de calidad). Dado un conjunto de validación el cual contenga N muestras. Denotemos por TP (True Positives) al número de muestras positivas correctamente clasificadas y por FP (False Positives) al número de muestras negativas clasificadas como positivas. Sea además denotado TN (True Negative) y FN (False Negative) al conjunto de muestras negativas correctamente

		Valor Calculado	
		Positivo	Negativo
Predecir Clase	Positivo	Verdadero Positivo	Falso Negativo
	Negativo	Falso Positivo	Verdadero Negativo

Figura 2.1: Matriz de confusión

e incorrectamente clasificadas respectivamente. En la Tabla 2.1 se describen algunas de las medidas de calidad mayormente utilizadas (Shulcloper y Vega, 2012).

- **Precisión** mide el porcentaje de muestras correctamente clasificadas.

$$\frac{TP + TN}{N} \quad (2.1)$$

- **Especificidad** mide la proporción de muestras negativas correctamente clasificadas.

$$\frac{TN}{TN + FP} \quad (2.2)$$

- **Sensibilidad** mide la proporción de muestras correctamente clasificadas.

$$\frac{TP}{TP + FN} \quad (2.3)$$

- **Valor de predicción positiva** da una medida del porcentaje de los verdaderos positivos de total de muestras clasificadas como positivas. Mide la probabilidad de que una muestra clasificado sea verdaderamente positiva.

$$\frac{TP}{TP + FP} \quad (2.4)$$

- **Valor de predicción negativa** da una medida del porcentaje de los verdaderos negativos de total de muestras clasificadas como negativas. Mide la probabilidad de que una muestra clasificada como negativa sea verdaderamente negativa.

$$\frac{TN}{FN + TN} \quad (2.5)$$

## 2.2. Algoritmos de clasificación

### 2.2.1. Algoritmo Random Forest

Random Forest (Bosques aleatorios), esta es una técnica que mejora la precisión de la clasificación incorporando aleatoriedad en la construcción de clasificadores individuales del tipo de árbol de decisión. También hay aleatoriedad en el arranque de los datos para cada árbol de decisión, y hay aleatoriedad en la forma en que se elige cada dimensión. El siguiente algoritmo resume el funcionamiento interno del bosque aleatorio.

**Datos:** conjunto de datos  $T = (x, y)$ , número de árboles  $m$ , número de niveles aleatorios  $k$

**Resultado:** RF, un conjunto de árboles adultos

Inicialización RF

**para**  $i = 1$  a  $m$  **hacer**

$T' \leftarrow$  escuchador( $T$ )  
    Arbol  $\leftarrow$  entrenamiento DT ( $T'k$ )  
    agregar árbol a RF

**fin**

#### Algoritmo 1: Random Forest

En esta técnica, los datos son los datos ( $T$ ) que son grabaciones previas ( $x$ ) con su diagnóstico correspondiente ( $y$ ), el número de árboles de decisión a generar ( $m$ ) y el número máximo de niveles que tendrá cada árbol ( $k$ ). Para cada árbol, se crea un nuevo conjunto ( $T'$ ) de datos de entrenamiento mediante el arranque (es decir, muestreo con reemplazo). Una vez que se seleccionan los datos, se selecciona un árbol seleccionando aleatoriamente dos características y eligiendo la mejor partición entre ambas características, esto se repite hasta que todas las características hayan sido analizadas o alcance el número máximo de niveles (Breiman, 2001).

### 2.2.2. Algoritmo Support Vector Machine

Support Vector Machine (SVM), es un modelo de clasificación basado en minimizar el error cuadrático de la clasificación, construyendo un hiperplano que separa los datos de la forma más precisa posible. SVM es considerado uno de los modelos más precisos y robustos posibles dentro de los algoritmos de clasificación binaria, siendo ampliamente utilizado por su versatilidad y efectividad a la hora de clasificar. Esto último se logra principalmente porque incluye la minimización del error estructural al clasificar nuevos objetos, que está directamente relacionado con uno de los principales objetivos de un clasificador, que es tener la habilidad de generalizar de forma correcta.

El método SVM define un hiperplano en  $\mathbb{R}^M$ , donde  $M$  es la cantidad de atributos, que separa lo mejor posible una clase de la otra. El objetivo es maximizar la distancia entre el hiperplano óptimo y los hiperplanos canónicos (que representan los bordes de cada clase). Para lograr esto se minimiza la norma Euclidiana de  $\mathbf{w}$ , que corresponde a los coeficientes que

definen el hiperplano, dando origen al siguiente problema de minimización como formulación primal. (Ferreira, 2012; Flores, Maldonado, y Weber, 2015):

$$\begin{aligned}
& \min_{w,b,\epsilon} \|w\|^2 + C \sum_{i=1}^N \xi_i \\
& \text{s.a. } y_i \cdot (W^T \cdot x_i + b) \geq 1 - \epsilon_i, i = 1, \dots, N, \\
& \quad \epsilon_i \geq 0, i = 1, \dots, N,
\end{aligned} \tag{2.6}$$

donde  $x_i \in \mathbb{R}^M$ ,  $y_i \in \{-1, 1\}$ ,  $\xi_i (i = 1, \dots, N)$  son variables de holgura que tienen por objetivo relajar las restricciones, permitiendo que ocurran errores, pero penalizándolos en la función objetivo. Esta penalización se controla con un parámetro  $C$ . La formulación previa puede ser extendida a clasificadores no lineales, usando el *kernel trick*: Los datos de entrenamiento son transformados en un espacio de mayor dimensionalidad  $\mathcal{H}$  a través de una función  $\phi : x \rightarrow \phi(x) \in \mathcal{H}$  de

en un producto interno en el espacio de  $\mathcal{H}$ , lo que nos lleva a la siguiente formulación (luego de calcular el dual):

$$\begin{aligned}
& \max_{\alpha} \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,s=1}^N \alpha_i \alpha_s y_i y_s K(x_i, x_s) \\
& \text{s.a. } \sum_{i=1}^N \alpha_i y_i = 0, \\
& \quad 0 \leq \alpha_i \leq C, i = 1, \dots, N.
\end{aligned} \tag{2.7}$$

### 2.2.3. Algoritmo Adaboost

El algoritmo Adaboost o Boosting adaptativo introducido en es un meta-algoritmo (un procedimiento que usa otro procedimiento como subrutina) que toma un conjunto de muestras etiquetadas  $\sum$  una distribución discreta  $D$  y un aprendiz débil *Weak* para retornar un clasificador combinado en  $T$  iteraciones. En cada iteración  $t$  Adaboost ejecuta *Weak* sobre el conjunto con la distribución  $D_t$  para obtener la hipótesis  $h_t$ . De acuerdo al desempeño de  $h_t$ , el algoritmo modifica  $D_t$  dándole menor peso a las muestras bien clasificadas y mayor peso a las muestras mal clasificadas con el objetivo que el siguiente clasificador se concentre en estas últimas, maximizando la cantidad de información que obtendrá en la siguiente ronda (García y Lozano, 2007).

**Entrada:**  $S = \{x_i, y_i\}_{i=1}^m, D_1, T, Weak(\cdot, \cdot)$

**Salida:**  $H(z)$

para  $t=1$  hacer

**Obtener una hipótesis débil usado**  $D_i: h_1 \leftarrow Weak(S, D_1)$   
    **Escoger adecuadamente**  $\alpha_t \in R$  **usualmente**  $\alpha_t = \frac{1}{2} \ln \left( \frac{1 - R_{emp}(h_t, S, D_t)}{R_{emp}(h_t, S, D_t)} \right)$   
    **Actualizar**  $D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_t h(x_i))}{Z_t}$   
    **Donde**  $Z_t$  **es un factor de normalización escogido para que**  $D_{t+1}$  **sea una distribución**  
**fin**

**Salida:**  $H(x) = \text{sgn} \left( \sum_{t=1}^T \frac{\alpha_t}{\sum_t \alpha_t} h_t(x) \right)$

**Algoritmo 2:** Adaboost

## 2.2.4. Algoritmo LogitBoost

LogitBoost es un algoritmo de impulso. Esto extiende AdaBoosts con un marco probabilístico en el cual la pérdida exponencial es reemplazada por una probabilidad condicional de Bernoulli. Podemos pensar en LogitBoosts como una regresión aditiva en la que hay tres componentes principales: la pérdida, la función del modelo y el algoritmo de optimización. La pérdida es un LogitBoost, esta es una regresión logística, el modelo es árboles de decisión aditivos. El siguiente algoritmo muestra el funcionamiento interno de LogitBoost (Otero y Sánchez, 2006; Ferreira, 2012).

**Entrada:** Conjunto de datos  $Z = \{z_1, z_2, \dots, z_n\}$  con  $z_i = (x_i, y_i)$ , donde  $x_i \in \mathcal{X}$  y  $y_i \in \{-1, +1\}$   $M$ , el número máximo de clasificadores.

**Salida:** Un clasificador  $H : \mathcal{X} \rightarrow \{-1, +1\}$

Inicializa los pesos  $w_1 = 1/N, i \in 1, \dots, N$

; **para**  $m=1$  a  $M$  **y mientras**  $H_m \neq 0$ ; **hacer**

    a) calcular los pesos y la respuesta de trabajo  $z_i = \frac{y_i^* - p(x_i)}{p(x_i)(1-p(x_i))}$  y pesas  $w_i = np(p(x_i)(1-p(x_i)))$ .  
    b) Ajustar la función  $H_m(x)$ , mediante una regresión ponderada de mínimos cuadrados de  $z_i$  a  $x_i$ , con pesos  $w_i$   
    c) Conjunto  $H_m(X) = H(x) + \frac{1}{2} H_m(x)$  and  $p(x) = \frac{\exp(H(x))}{\exp(H(x)) + \exp(-H(x))}$

**fin**

Clasificador final:  $H(x) = \text{sign} \left( \sum_{j=1}^M \alpha_j H_j(x) \right)$

**Algoritmo 3:** LogitBoost

## 2.3. Herramientas computacionales

### 2.3.1. Hardware

Las herramientas para el hardware se utilizado en los experimentos tiene las siguientes especificaciones: Computadora DELL inspiron 14 intel CORE i3, memoria Ram de 6 gb, disco duro 1TB, sistema operativo Windows 10.

- **Arduino Mega:** Arduino MEGA 2560 está diseñado para proyectos que requieren más líneas I / O, más memoria de croquis y más RAM. Con 54 pines de E / S digitales, 16 entradas analógicas y un espacio más grande para su boceto, es la placa recomendada para impresoras 3D y proyectos de robótica. Esto le da a sus proyectos mucho espacio y oportunidades manteniendo la simplicidad y efectividad de la plataforma Arduino <sup>2</sup>.
- **Sensor MPU6050:** es el motor interno Digital Motion Processing es compatible con 3D MotionProcessing y gestos algoritmos de reconocimiento El MPU-60500 recoge datos de giroscopio y acelerómetro al sincronizar el muestreo de datos en un tasa definida por el usuario. El conjunto de datos total obtenido por el MPU-60X0 incluye datos de giroscopio de 3 ejes, 3-Datos del acelerómetro del eje y datos de temperatura. La salida calculada de la MPU para el sistema el procesador también puede incluir datos de rumbo de un magnetómetro digital de 3 ejes de un tercero. El FIFO almacena el conjunto completo de datos, reduciendo los requisitos de tiempo en el procesador del sistema permitiendo que la ráfaga del procesador lea los datos FIFO. Después de reventar los datos FIFO, el sistema el procesador puede ahorrar energía ingresando a un modo de reposo de baja potencia mientras la MPU recolecta más datos <sup>3</sup>.
- **Sensor ADXL345:** el ADXL345 es un acelerómetro de 3 ejes pequeño, delgado y de baja potencia con una medición de alta resolución (13 bits) de hasta 16 g. Digital los datos de salida están formateados como complemento a dos dígitos de 16 bits y es accesible a través de un SPI (3 o 4 hilos) o I2 C interfaz digital. El ADXL345 es ideal para aplicaciones de dispositivos móviles. Eso mide la aceleración estática de la gravedad en aplicaciones de detección de inclinación, así como la aceleración dinámica resultante del movimiento o shock. Su alta resolución (4 mg / LSB) permite la medición de inclinación cambia menos de 1.0.Se proporcionan varias funciones de detección especiales. Actividad y la detección de inactividad detecta la presencia o la falta de movimiento y si la aceleración en cualquier eje excede un nivel establecido por el usuario. Detección de Tap detecta grifos simples y dobles. La detección de caída libre detecta si el dispositivo está cayendo Estas funciones se pueden mapear a uno de dos interrumpir los pines de salida. Un nivel integrado de patente pendiente de 32 niveles. El buffer primero en entrar, primero en salir (FIFO) se puede usar para almacenar datos en minimizar la intervención del procesador host <sup>4</sup>.

---

<sup>2</sup><https://www.arduino.cc/en/Guide/ArduinoMega2560>

<sup>3</sup>[https://store.invensense.com/datasheets/invensense/MPU-6050\\_DataSheet\\_V3204.pdf](https://store.invensense.com/datasheets/invensense/MPU-6050_DataSheet_V3204.pdf)

<sup>4</sup><https://www.sparkfun.com/datasheets/Sensors/Accelerometer/ADXL345.pdf>

### 2.3.2. Software

1. **Python** es un lenguaje de programación poderoso y fácil de aprender. Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. La elegante sintaxis de Python y su tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas <sup>5</sup>.
2. **Anaconda** es una distribución gratuita y de código abierto [5] de los lenguajes de programación Python y R para aplicaciones relacionadas con la ciencia de datos y el aprendizaje automático (procesamiento de datos a gran escala, análisis predictivo , informática científica ), que pretende simplificar la administración e implementación de paquetes .Las versiones del paquete son administradas por la conda del sistema de gestión de paquetes .La distribución de Anaconda es utilizada por más de 6 millones de usuarios e incluye más de 250 populares paquetes de ciencia de datos adecuados para Windows, Linux y MacOS <sup>6</sup>.
3. **Weka** (Waikato Environment for Knowledge Analysis, en español «entorno para análisis del conocimiento de la Universidad de Waikato») es una plataforma de software para el aprendizaje automático y la minería de datos escrito en Java y desarrollado en la Universidad de Waikato. WEKA es software libre distribuido bajo la licencia GNU-GPL <sup>7</sup>.
4. **Arduino** es una compañía open source de hardware y software, así como un proyecto y comunidad internacional que diseña y manufactura placas de desarrollo de hardware para construir dispositivos digitales y dispositivos interactivos que puedan censar y controlar objetos del mundo real. Arduino se enfoca en acercar y facilitar el uso de la electrónica y programación de sistemas embebidos en proyectos multidisciplinarios. Los productos que vende la compañía son distribuidos como Hardware y Software Libre, bajo la Licencia Pública General Reducida de GNU (LGPL) o la licencia pública general de GNU (GPL) <sup>8</sup>.
5. **Excel** es una aplicación de hojas de cálculo que forma parte de la suite de oficina Microsoft Office. Es una aplicación utilizada en tareas financieras y contables, con fórmulas, gráficos y un lenguaje de programación.
6. **Matlab** (abreviatura de MATrix LABoratory, “laboratorio de matrices”) es una herramienta de software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Está disponible para las plataformas Unix, Windows, Mac OS X y GNU/Linux, entre sus prestaciones básicas se

---

<sup>5</sup><https://www.python.org/>

<sup>6</sup><https://anaconda.org/anaconda/python>

<sup>7</sup><https://www.cs.waikato.ac.nz/ml/weka/>

<sup>8</sup><https://www.arduino.cc/en/Guide/Introduction>

hallan: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes y con otros dispositivos hardware <sup>9</sup>.

7. **Scikit-learn** (anteriormente scikits.learn ) es una biblioteca de aprendizaje de máquina de software libre para el lenguaje de programación Python. Presenta varios algoritmos de clasificación , regresión y agrupación , incluyendo máquinas de vectores de soporte , bosques aleatorios , aumento de gradiente , k- medias y DBSCAN , y está diseñado para inter-operar con las bibliotecas numéricas y científicas de Python, NumPy y SciPy <sup>10</sup>.

---

<sup>9</sup><https://www.mathworks.com/products/matlab.html>

<sup>10</sup><http://scikit-learn.org/>

## **Capítulo 3**

### **Estado del arte**

En el estado del arte se divide en dos secciones en primera sección muestra trabajos relacionados con la red de sensores y en el segunda sección con la clasificación de ND.

### 3.1. Trabajos relacionados con la Red de sensores

Mannini *et al.*, la investigación de la ataxia de inicio temprano (EOA) y el trastorno de coordinación del desarrollo (DCD) son dos condiciones que afectan la coordinación en los niños. La identificación fenotípica de la coordinación alterada juega un papel importante en su diagnóstico. La marcha es una de las pruebas incluidas en las escalas de calificación que se pueden usar para evaluar la coordinación motora. Un problema práctico es que la semejanza entre los síntomas EOA y DCD puede dificultar su diagnóstico. En este estudio empleamos sensores inerciales y un clasificador supervisado para obtener una clasificación automática de la condición de los participantes. Los datos de las unidades de medición inercial montadas en la caña y la cintura se usaron para extraer características durante la marcha en niños diagnosticados con EOA o DCD y emparejados por edad controles (Mannini y cols., 2017).

Bregou *et al.*, se investiga el entorno de andar natural de un niño puede verse afectado por el entorno del laboratorio de la marcha. Se han desarrollado dispositivos portátiles que usan sensores corporales para el análisis de la marcha. El objetivo de este estudio fue validar y explorar el uso de sensores inerciales de uso en el pie para la medición de los parámetros espacio-temporales seleccionados, basados en la trayectoria del pie 3D, en niños que caminan independientemente con parálisis cerebral (PC). Realizamos un estudio de casos y controles con 14 niños con PC de 6 a 15 años de edad y 15 controles de la misma edad. La precisión y la precisión del dispositivo de pie se midieron utilizando un sistema óptico de captura de movimiento como sistema de referencia. Los niños con parálisis cerebral mostraron una variabilidad inter-zancada significativamente mayor (medida por su coeficiente de variación) para la velocidad, la longitud de la zancada, el balanceo y la postura (Bourgeois, Mariani, Aminian, Zambelli, y Newman, 2014).

Rosenbaum *et al.*, existe diferentes investigaciones y se ha demostrado que la velocidad de la marcha influye en los patrones de carga del pie en adultos, pero el mecanismo no se ha investigado en niños. El presente estudio investigó los efectos de los cambios en la velocidad de la marcha sobre las características de carga del pie en 20 niños con desarrollo típico que participaron en mediciones de presión plantar a velocidades de marcha lenta, normal y rápida. A pesar de los tiempos de contacto más cortos en la condición de velocidad de marcha rápida, se observó un aumento significativo de la carga del pie en el ante pié, el ante pié y los dedos media y central del pie, mientras que disminuyó ligeramente en el medio pié y el ante pié laterales. Los resultados generalmente confirman los hallazgos en adultos de que la velocidad de la marcha no afecta uniformemente las características de carga del pie y que estos efectos deben tenerse en cuenta al comparar diferentes grupos de sujetos o niños en ocasiones de medición repetidas (Rosenbaum, Westhues, y Bosch, 2013).

Shiqi Yu *et al.*, en investigaciones del reconocimiento de la marcha ha demostrado ser útil en la identificación humana a distancia. Sin embargo, muchas variaciones, como la vista, la ropa y el estado de transporte, hacen que el reconocimiento de la marcha siga siendo un desafío en las aplicaciones reales. Las variaciones hacen que sea difícil extraer la característica invariable para distinguir diferentes temas. Para la variación de vista, se puede emplear un modelo de transformación de vista para convertir la característica de marcha de una vista a otra. La mayoría de los modelos existentes primero necesitan estimar el ángulo de visión, y pueden funcionar solo para un par de vistas. No pueden convertir datos de vistas múltiples a una vista específica de manera eficiente. Otras variaciones también necesitan algunos modelos específicos para manejar. Empleamos un modelo profundo basado en autocoder para la extracción invariante de la marcha. El modelo puede sintetizar la función de marcha de forma progresiva mediante autocoders multicapa apilados. La ventaja única es que puede extraer la función de marcha invariante utilizando solo un modelo, y la característica extraída es robusta para ver, vestir y variar la condición de transporte. El método propuesto se evalúa en dos grandes conjuntos de datos de la marcha, CASIA Gait Dataset B y SZU RGB-D Gait Dataset. Los resultados experimentales muestran que el método propuesto puede lograr el rendimiento más avanzado con un solo modelo uniforme (Yu, Chen, Wang, Shen, y Huang, 2017).

Cain *et al.*, en investigaciones del tema de los niños con obesidad se mueven menos y con mayor dificultad que sus contra partes de peso normal. los objetivos de este estudio fueron caracterizar la calidad de movimiento de los niños que realizan la prueba de estado de múltiples etapas (MSFT) e informar cómo las características de calidad de movimiento se agrupan según el estado del peso. La RHO de SPEARMAN se utilizó para evaluar la relación con el grupo de IMC, y se utilizó una prueba de U de Mann-Whitney para evaluar las diferencias entre los grupos de IMC. El análisis adicional reveló características de rendimiento clave que difieren entre los grupos de IMC. Estos fueron (i) representativos del rendimiento de los niños durante el MSFT y (ii) se correlacionaron significativamente de forma negativa con el IMC (Clark, Barnes, Holton, Summers, y Stratton, 2016).

Nursuriati *et al.*, la investigación de los niños con Trastorno del Espectro Autista (TEA) en Malasia, se realizan muchos esfuerzos y estudios para comprender y manejar los asuntos relacionados con el TEA. Una forma es encontrar los medios para facilitar las comunicaciones sociales entre estos niños y sus cuidadores, particularmente durante la intervención. Si el cuidador es capaz de comprender el estado emocional de los niños antes de la terapia, se desarrollará algún tipo de confianza y apego. Sin embargo, regular las emociones es un desafío para estos niños. La comunicación no verbal, como la expresión facial, es difícil para los niños con TEA. Por lo tanto, propusimos el uso de patrones de caminar (es decir, marcha) para detectar el tipo de emociones de los niños con TEA. Aunque el uso de la marcha para el reconocimiento de emociones es común entre las personas normales, ninguno se puede encontrar hecho en niños con TEA. Por lo tanto, el objetivo de este trabajo es realizar una revisión preliminar sobre las posibilidades de llevar a cabo la detección de emoción basada

en la marcha entre los niños con TEA con respecto a los tipos emocionales, los parámetros de la marcha y los métodos de adquisición de datos de la marcha (Jamil, Khir, Ismail, y Razak, 2015).

Matthew *et al.*, muy pocos estudios han examinado los patrones de la marcha de los niños con autismo. Una mayor conciencia de las desviaciones de movimiento podría ser beneficiosa para la planificación del tratamiento. El propósito de este estudio fue comparar los patrones cinemáticos y cinéticos de la marcha en niños con autismo versus controles de la misma edad. Métodos: Doce niños con autismo y veintidós controles de la misma edad participaron en el estudio. Se utilizó un sistema de captura de movimiento de ocho cámaras y cuatro placas de fuerza para calcular los ángulos de las articulaciones y la cinética de las articulaciones durante la marcha. Los análisis paramétricos y los análisis del componente principal se aplicaron a las variables de forma de onda cinética y cinética de los grupos de autismo ( $n = 12$ ) y control ( $n = 22$ ). Las diferencias grupales en los valores de parametrización y las puntuaciones de los componentes principales se analizaron mediante ANOVA de un factor y pruebas de Kruskal-Wallis. Los resultados sugieren que los niños con autismo demuestran momentos de flexión plantar reducidos y ángulos de dorsiflexión aumentados, que pueden estar asociados con la hipotonía (Calhoun, Longworth, y Chester, 2011).

### **3.2. Trabajos relacionados con la clase de ND**

Ioannis *et al.*, diferentes métodos se muestra mediante la obtención de registro de información electrónica de pacientes, se realizó la revisión sistemática en las aplicaciones de machine learning con el aprendizaje supervisado, utilización minería de datos y técnicas en el área de la diabetes para prevención y diagnóstico al final se realizó un esfuerzo sistemático para identificar los enfoques del aprendizaje de máquina opinión y la minería de datos aplicada a la investigación en la diabetes mellitus (Kavakiotis y cols., 2017).

Patterson *et al.*, propuso detectar las diferentes condiciones de una persona; normal y rígido en el tobillo con un acelerómetro para detectar la marcha del paciente y obtener datos, se utilizó un sistema captura de movimiento CODA (Charnwood Dynamics, Leicestershire, Reino Unido) se empleó para recopilar datos cinemáticos y como resultado los patrones cinemáticos de la marcha alteraron los cambios en el movimientos de tobillo y el estudio realizado puede ser utilizado con personas de enfermedad Parkinson o diabetes mellitus .

Gomes *et al.*, la investigación con pacientes que padecen ND han sufrido disturbios que han sido relacionados experimentalmente con ulceraciones plantares que padecen. El objetivo es simular mediante un patrón de la marcha como el sistema neuromusculo-esquelético del diabético neuropático trata con un musculo distal reducido durante la marcha nivelada. El modelo plano musculo-esquelético adoptado en el estudio, fue capaz de representar la estrategia de cadera posiblemente adoptada por la función utilizada que da lugar a la

inestabilidad de la marcha .

L. H. Anjaneya *et al.*, Se investigo los riesgos de la diabetes en aumentando en niños y adultos. Se han propuesto diversos enfoques para la detección temprana de la diabetes y la prevención en ella el objetivo es utilizar señales EMG (señales electromiografías) para la clasificación de la diabetes, debido a los artefactos de movimiento en las señales EMG y clasificar la señal de manera eficiente un nuevo método mediante la consideración del dominio del tiempo y de la frecuencia de características de las señales EMG. Se obtuvo la etapa de preprocesamiento, eliminación de ruido o normalización de datos, se extraen características posteriores de la señal dada mediante a redes neuronales .

Malindu Fernando *et al.*, Se realizaron búsquedas sistemáticas en bases de datos electrónicas de artículos que informan sobre el efecto de la neuropatía periférica diabética sobre la marcha, la electromiografía dinámica y las presiones plantares, los resultados de los análisis marinos sugirieron un tiempo de postura más largo y moderadamente mayor plantar en pacientes diabéticos de neuropatía periférica en el pie trasero, el pie medio y el ante pie en comparación con los controles. La revisión sistemática de los estudios sugirió posibles diferencias en las características biomecánicas de los pacientes con neuropatía diabética, sin embargo, estos hallazgos fueron inconsistentes y limitados .

Haitham M. Al-Angari *et al.*, este trabajo, presentaron nuevas características de forma que capturan las variaciones en la presión plantar usando medidas de forma y entropía para el estudio de pacientes con retinopatía, DPN y nefropatía, y un grupo de control diabético sin complicaciones. El cambio en la posición de la presión pico de la planta (PPP) con cada paso para ambos pies se representó como un polígono convexo, índice de asimetría, área del polígono convexo, segundo momento wavelet (WM2) y entropía de la muestra (SamEn).

Berki *et al.*, En este estudio, se recogieron datos de presión y tensión de cortante de 26 sujetos diabéticos y control en total usando una nueva instrumentación que mide los vectores de fuerza vertical y horizontal de la superficie de contacto plantar durante múltiples instancias en el ciclo de la marcha. Se realizó una Transformada de Fourier Discreta bidimensional en cada conjunto de datos, para cada uno de los diez tamaños de sensor. Los resultados mostraron que un sensor que mide 9,6 mm por 9,6 mm causó reducciones significativas en los tres componentes de tensión ( $p < 0,001$ ), mientras que los sensores que miden 1,6 mm por 1,6 mm, hasta 4,8 mm por 4,8 mm, pueden capturar toda la gama espacial frecuencias en los datos de presión y tensión de corte .

Marcela R. Camargo *et al.*, el estudio fueron evaluar aspectos del equilibrio, la fuerza del tobillo y los parámetros de la marcha espaciotemporal en individuos con neuropatía periférica diabética (NPD) y verificar si los déficits en los parámetros de la marcha espaciotemporal se asociaron con la fuerza muscular y el equilibrio del tobillo. Los parámetros espaciotemporales de la marcha se evaluaron midiendo el tiempo de caminata para establecer la distancia

durante las velocidades de caminata autoelegidas y máximas. La movilidad funcional y el rendimiento del equilibrio se evaluaron utilizando las pruebas de Alcance funcional y Time Up and Go. La fuerza muscular isométrica del tobillo se evaluó con un dinamómetro digital portátil. Se emplearon análisis de varianza para verificar posibles diferencias entre grupos y condiciones. Se empleó un análisis de regresión lineal múltiple para descubrir posibles predictores de déficits de la marcha. a movilidad espaciotemporal, la movilidad funcional, el rendimiento del equilibrio y la fuerza muscular del tobillo se vieron afectados en individuos con NDP. El rendimiento de la prueba Time Up and Go y la fuerza isométrica del músculo del tobillo se asociaron a los cambios de marcha espaciotemporal, especialmente durante la condición de velocidad de marcha máxima.

Sánchez-Delacruz *et al.*, se propuso un modelo en el que la gente está rodeada por un entorno digital consciente de su presencia que responde de manera adaptativa a las necesidades y hábitos de los usuarios para facilitar la vida diariamente en casa, ocio y trabajadora asistencia sanitaria es una. El tema tratado en son las habilidades motoras han sido estudiadas en la literatura algoritmos para una enfermedad específica. Se obtuvieron a través del mismo meta-clasificador (LogRS). Adicionalmente, el objetivo fue encontrar que LogRS claramente clasificar entre healthy y los enfermos, esto es importante que los meta-clasificadores están bien posicionados para enfermedades neurodegenerativas clasificar diferentes enfermedades sin tener que adaptarse a algoritmos.

Hastings *et al.*, el objetivo del estudio fueron determinar los predictores de alineación y función de la columna medial en personas con diabetes y neuropatía periférica. Métodos: 23 participantes con diabetes y neuropatía recibieron radiografías, cinemática de aumento de talón, resonancia magnética y pruebas musculares isocinéticas para medir: alineación de la columna medial (ángulo de Meary: ángulo entre el eje longitudinal del primer metatarsiano y la cabeza y el cuello del astrágalo), función de la columna medial (antepié con respecto a la flexión plantar del retropié durante el aumento del talón), volumen del tejido muscular y grasa intrínsecos, relación entre el volumen tibial posterior del tibial y el tendón flexor digitorum, función de la fascia plantar (cambio de ángulo de Meary de los dedos planos a extendidos) par máximo de flexión plantar. El deterioro de las estructuras de soporte de la columna medial predijo la alineación y la función. Se requiere una investigación prospectiva para monitorear la alineación, la estructura y la función a lo largo del tiempo para informar las estrategias de intervención temprana para prevenir la deformidad, la ulceración y la amputación .

# Capítulo 4

## Metodología

Las pruebas médicas realizadas a cada paciente para el diagnóstico de neuropatía diabética se miden a partir de la parestesias, (fuerza muscular, reflejos de tobillo, sensación de vibración). Sin embargo no todos los métodos utilizados por los médicos se obtiene correctamente el nivel neuropatía diabética.

Por lo anterior, se propone el diseño de un modelo de clasificación de neuropatía diabética. Para lograr el objetivo se llevaron a cabo las siguientes etapas.

#### 1. **Recolección de datos:**

- Construcción de la red de sensores para la captura de biomarcadores de la marcha.
- Buscar una institución y medico que brinde atención a pacientes con diabetes y que sufren neuropatía diabética.
- Investigar en diversas fuentes de salud cuales son las mas variables determinadas como factores de riesgo en el diagnóstico de la neuropatía diabética, además de solicitar la ayuda de un experto quien se encarga de validar y agregar si es necesario variables a partir de la hoja ingreso que se realiza al asistir a la consulta.

#### 2. **Construcción de la base de datos:**

- El paciente realiza una serie de caminatas durante un tiempo de 5 minutos, el cual el experto determinar si el pacientes se encuentra sano o sufre neuropatía diabética.
- Realizar la construcción de la base de datos con los registros obtenidos por los pacientes de la Servicio médico de la Universidad Juárez Autónoma de Tabasco.
- Analizar la información obtenida y buscar trabajos relacionado en el estado del arte que sean de apoyo a seleccionar los algoritmos o técnicas para la solución del problema planteado.

#### 3. **Preprocesamiento de datos:**

En la preparación de los datos, manipulación y transformación de la información sin refinar y contenida en el conjunto de datos se puedan ser descubierta o estar accesible de forma más fácil. El proceso consiste en completar datos perdidos, disminuir el ruido, eliminar valores atípicos y resolver inconsistencia de los datos, además de tratar la redundancia de los mismos.

#### 4. **Predicción de modelo**

- Investigar y analizar los algoritmos de aprendizaje automático y seleccionar el mas óptimo a utilizar como clasificador en el modelo.

#### 5. **Diagnóstico**

- Se comparan los algoritmos de clasificación, se determinar cual es el mas óptimo para dar un diagnóstico, por paciente, segmento y se muestra la matriz de confusión.

## **Capítulo 5**

# **Experimentos y resultados**

El presente trabajo se propone la construcción de una red de sensores y un modelo de clasificación relacionado a través de los patrones de la marcha y sea de apoyo al médico en el diagnóstico. Se observa la Figura 5.1, en el lado del usuario (es decir, paciente) la red de los sensores (es decir, el dispositivo) es llevada por él o ella. Consiste en cinco acelerómetros, Arduino y un arnés, son ligeros y cómodos de usar por medio de ser montados en un arnés elástico que podría ajustarse fácilmente en varios tipos de usuarios. Las mediciones de la red de sensores se guardan en una base de datos por paciente determinada la ubicación del archivo. En la caminata del paciente se sugiere caminar en una línea recta hasta cierto punto, vuelta y reaparición. Los datos recolectados serán accedidos por un experto que realizará un análisis de los datos y producirá un diagnóstico.

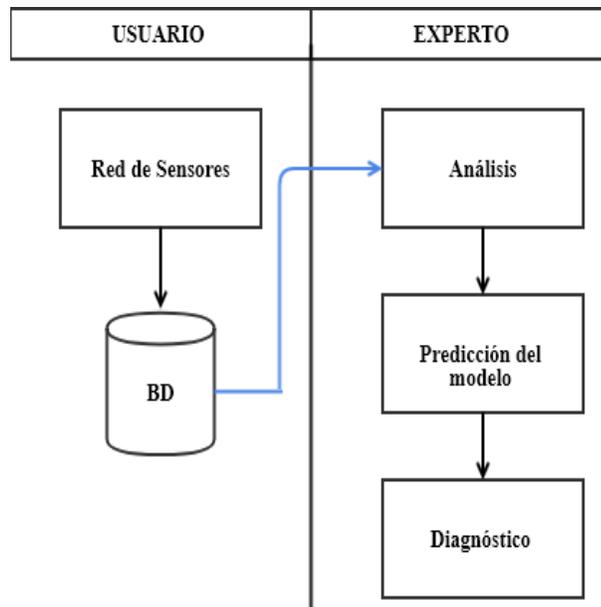


Figura 5.1: Figura proceso de clasificación.

## 5.1. Experimento I

En esta primera fase se presenta el diseño de una red de sensores accesible para registrar la marcha y correspondiente al aprendizaje automático con el fin de crear un modelo predictivo de la misma. El dispositivo está dirigido a ayudar a los expertos en el diagnóstico de alguna enfermedad relacionada con la marcha. Sin embargo, en la etapa de la investigación proponemos utilizar la información del sensor para discriminar entre géneros ya que se ha estudiado antes.

### 5.1.1. Construcción del Sensor rodilla-tobillo

Se propone la construcción de una pequeña red sensores utilizando los materiales que se muestran en la Tabla 5.1, se enfatiza que la red de sensores sea de bajo costo y fácil de manejar para el experto. La topología de la red se muestra en la Figura 5.2, en donde la información fluye en las entradas SDA y SCL de una tarjeta arduino mega.

Material	Precio (en MXN)
Arduino Mega	\$ 300.00
Sensor MPU6050 (2)	\$ 160.00
Velcro	\$ 50.00
Elastic	\$ 40.00
Case	\$ 120.00
Dupont cables	\$ 100.00
<b>Total:\$ 670.00</b>	

Tabla 5.1: Materiales y costo del sensor rodilla-tobillo

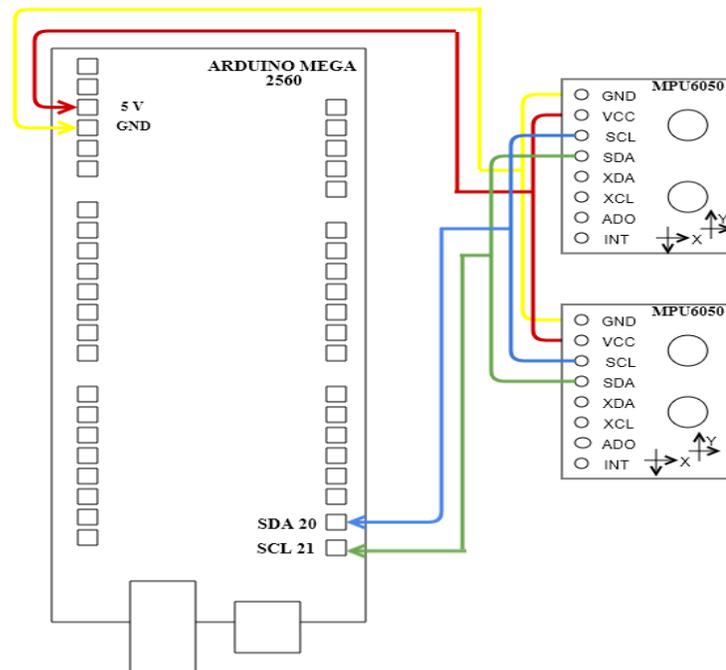


Figura 5.2: Topología del sensor rodilla/ tobillo.

### 5.1.2. Recolección de datos

El la recolección de datos se trabajó de la siguiente manera, los niños fueron orientados en lo que el experimento consistió y donde cada uno de los sensores fue localizados como se puede apreciar en la Figura 5.3. Una vez que la red de sensores fue colocada en el niño, una prueba fue realizada para identificar cualquier problema con el sensor o en el ajuste en la colocación de los sensores. Después de la prueba el niño fue ordenado a caminar en un área amplia, en este caso era la explanada de una escuela secundaria. Se le pidió al niño que caminara en línea recta, el tiempo de caminata se tomó aproximadamente 3 minutos y los datos fueron recolectados en un archivo .CSV para cada niño. Los registros obtenidos se unieron en un solo archivo .CSV para realizar experimentos con varios algoritmos de aprendizaje automático.



Figura 5.3: Colocación de la red sensores.

Para probar el dispositivo y la metodología propuesta de aprendizaje automático realizamos una tarea de identificación de género, con el objetivo de determinar el género de un niño basado en los datos de su marcha. Se Registraron 10 niños caminando y establecemos el Arduino para registrar 400 muestras en 3 minutos, que es la cantidad máxima de tiempo que los niños les llevó caminar la ruta predefinida. En la Tabla 5.2 se muestran los datos estadísticos de los diez niños que participaron en el experimento, los cuales son 5 niñas y 5 niños, el grupo tiene una altura media de 1,58 metros, peso medio de 57 kg y una masa corporal promedio de 21,4 . Los niños mayores tienen 15 años de edad, mientras que el más chico 12 años de edad. Su forma física es normal para su edad y etapa de desarrollo.

Genero	Edad	Estatura	Peso	Masa corporal
Niño	14	1.60 m	60 kg	23.43
Niño	14	1.65 m	61 kg	22.51
Niña	12	1.59 m	48 kg	19.22
Niña	13	1.62 m	56 kg	21.33
Niño	13	1.58 m	62 Kg	24.83
Niño	15	1.58 m	53 kg	21.23
Niño	14	1.65 m	61 kg	22.71
Niña	14	1.50 m	43 kg	19.28
Niña	15	1.54 m	47 kg	19.86
Niña	12	1.50 m	44 kg	19.55

Tabla 5.2: Información de sujetos de pruebas

### 5.1.3. Análisis de datos

Una vez obtenida la información de cada sensor en cada niño niños caminando, se procedió a realizar un análisis de datos utilizando el software *Weka*. Esta herramienta proporciona una gran colección de algoritmos, incluyendo *LogitBoost* y *RandomForest*, se realizaron varios experimentos con esta colección de métodos de aprendizaje automático. Dada la cantidad de datos utilizamos una configuración de validación cruzada en la que dejamos fuera 30 % de los datos para las pruebas. La evaluación de algoritmos se realizó con la configuración estándar del software **Weka**.

### 5.1.4. Resultados

Los resultados mostraron que al discriminar el género de los niños se obtiene una precisión de 92,04 %, lo que demuestra que los biomarcadores de la marcha y los algoritmos de aprendizaje automático *LogitBoost* y *RandomForest* son adecuadas para realizar esta tarea y además, para empezar a investigar otros aspectos como diagnóstico de otras enfermedades. La Figura 5.4 muestra las diferentes actuaciones de otras combinaciones de algoritmos y se observar el buen rendimiento que se obtiene, sin embargo hay pocos que no mejoran significativamente, en la línea de base que sería adivinar entre ambos sexos. Sin embargo, hay varios que tienen un rendimiento de más del 85 % que es bueno y señala que la tarea es de hecho discriminativo (Bernardino, Sánchez-DelaCruz, y Ruíz, 2018).

La Tabla 5.3 muestra la matriz de confusión de estos experimentos discriminativos. Se observa que hay pocos errores de confusión para los datos que recogimos, los datos de las niñas son las más confusas, pero tenemos más datos para esta clase, por lo que consideramos esto normal; esto es corroborado por el valor medio del espacio Roc: 0,972, lo que indica una alta tasa precisión de clasificación.

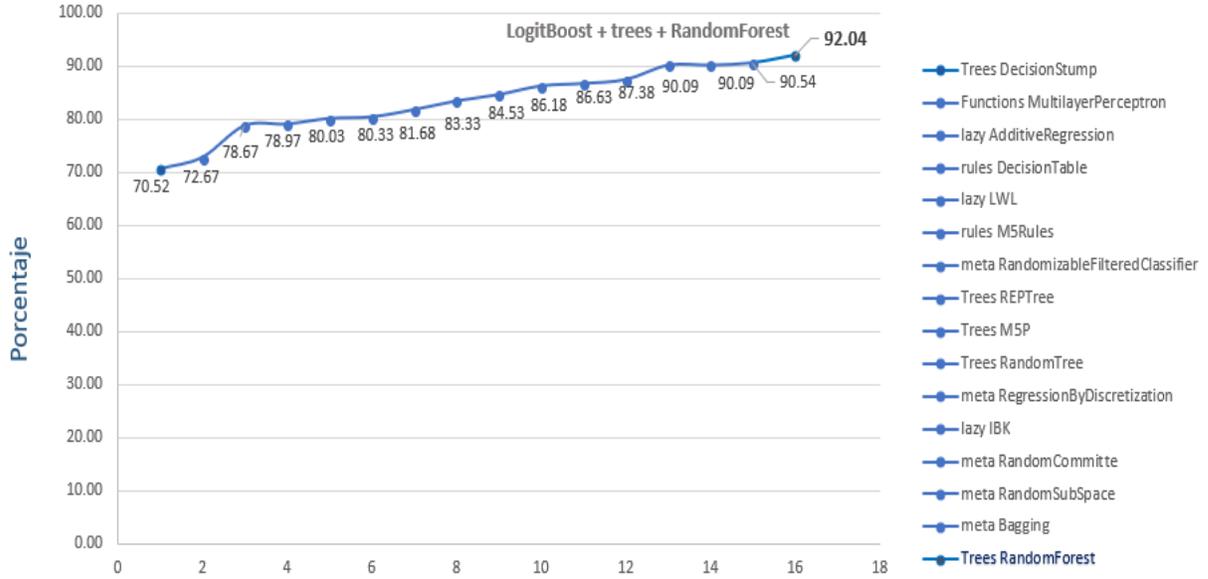


Figura 5.4: Algoritmos Metaclasificadores

a	b	class
282	35	a=niño
18	331	b=niña

Tabla 5.3: Matriz de confusión

## 5.2. Experimento II

Trabajando con el mismo diseño de experimento de la sección anterior, se adecuó para una segunda fase que consiste en experimentos para la clasificación de neuropatía diabética con datos obtenidos por una red sensores ubicados en la parte de de la espalda baja, rodilla derecha y rodilla izquierda, tobillo derecho y tobillo izquierdo.

### 5.2.1. Construcción de la red de sensores

Para la construcción de la red sensores se utilizaron los materiales que se aprecian en la siguiente Tabla 5.4. Para la conexión de cada sensor se diseño la topología de la Figura 5.5, y la información fluye en conexiones de manera analógica en el arduino mega, para almacenar la información se utilizaron los entornos de programación *Mallab* y *arduino*.

Material	Precio (en MXN)
Arduino Mega	\$ 300.00
Sensor ADLX335 (4)	\$ 320.00
Velcro	\$ 50.00
Elastic	\$ 40.00
Case	\$ 120.00
Dupont cables	\$ 100.00
<b>Total: \$830.00</b>	

Tabla 5.4: Materiales y costos

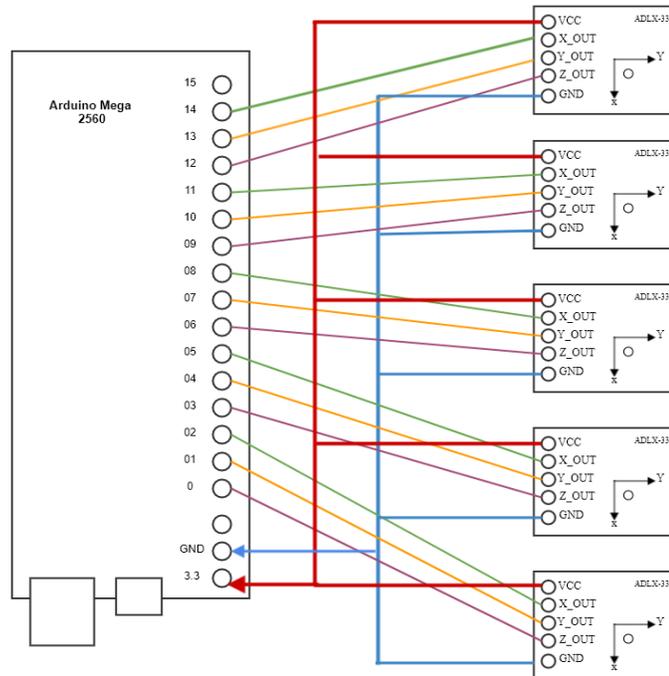


Figura 5.5: Topología de la red sensores.

### 5.2.2. Construcción del DataSet

El proceso de construcción de la base de datos se obtiene a partir de la marcha de pacientes con neuropatía diabética y sanos. El trabajo de recolección se llevó a cabo en la Unidad de servicios médicos de la Universidad Juárez Autónoma de Tabasco, Figura 5.7, bajo la supervisión del Dr. Roberto Weber Burque Cruz, el cual determinó si el paciente sufría de neuropatía diabética o era de la clase sujeto sano. En la recolección de datos se les explicó brevemente a los pacientes en que consistían las pruebas y en donde se encontraría colocado cada sensor (tobillo, rodilla y espalda baja), como se observa en la siguiente Figura 5.8. Los pacientes realizaron una caminata previa, la cual ayudó a los pacientes a adaptarse a la red

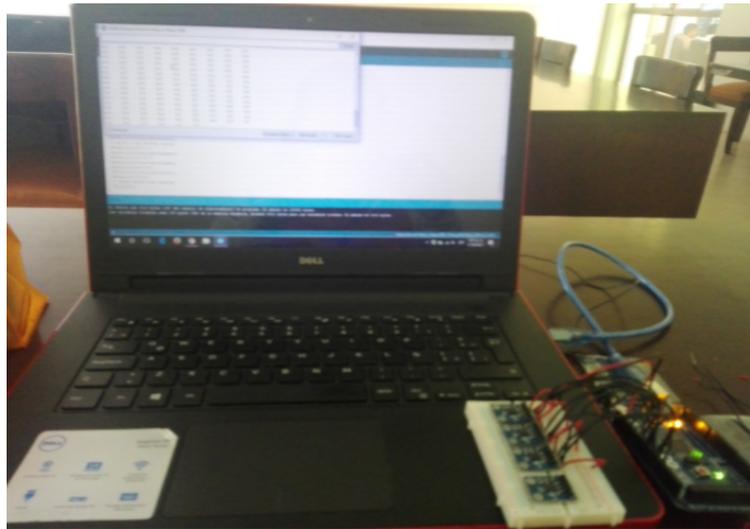


Figura 5.6: Prueba previas de sensores.

sensores o si sentía alguna molestia, cada uno de ellos caminó de manera lineal mientras que los datos era recabados sin interrupciones. La información obtenida fue almacenada en un archivo .csv y se configuro el dispositivo arduino para el registro de 400 puntos en 5 minutos la cantidad de datos en la ruta predefinida.

La Tabla 5.5 muestra a detalle los datos estadísticos de cada uno de los pacientes que participó en las pruebas. Se obtuvo un total de 10 pacientes con neuropatía diabética y 5 personas sanas.



Figura 5.7: Servicios médicos UJAT



Figura 5.8: Ubicación de la red sensores.

### 5.2.3. Estructura del dataset

Para una mejor manipulación y organización de los datos se almacenaron en un archivo separado por comas (.csv). Los archivos en formato CSV son documentos en formato abierto utilizados comúnmente para la representación de datos en forma de tabla, en este formato las columnas se separan por una coma (,) y las filas por un salto de línea. En la Figura 5.9 se visualiza un segmento del archivo .csv que contiene el conjunto de datos, el cual cada línea representa las tres posiciones obtenidas del sensor ocupado y se nombraron de la siguiente manera; espalda baja( $x,y,z$ ), rodilla derecha ( $x,y,z$ ), tobillo derecho ( $x,y,z$ ), rodilla izquierda ( $x,y,z$ ) y tobillo izquierdo ( $x,y,z$ ).

### 5.2.4. Preprocesamiento de datos

Los registros obtenidos de la red sensor, son los que conforman el conjunto de datos que se utilizaron para la elaboración del modelo, sin embargo presentan ciertas anomalías y es por ello que se debe realizar el preprocesamiento ya que esto puede mejorar el desempeño

Genero	Edad	Peso	Estatura	Tiempo de padecimiento	Causa
Masculino	54	89	1.70	5	Hereditario
Masculino	60	108	1.65	10	Alimentación
Femenino	56	99	1.60	4	Hereditario
Masculino	56	81.5	1.62	6	Hereditario
Masculino	62	73	1.57	15	Alimentación
Femenino	50	70	1.59	8	Hereditario
Masculino	58	102	1.61	6	Alimentación
Masculino	57	87.7	1.58	8	Alimentación
Femenino	61	90	1.65	3	Hereditario
Masculino	50	83.2	1.63	5	Hereditario
Femenino	35	72	1.61	- - - - -	Sano
Masculino	38	82	1.65	- - - - -	Sano
Masculino	45	95	1.67	- - - - -	Sano
Masculino	40	75	1.59	- - - - -	Sano
Femenino	29	59	1.55	- - - - -	Sano

Tabla 5.5: Información de pacientes.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	colz	coly	colx	rdizqz	rdizqy	rdizqx	rderz	rdery	rderx	tbizqz	tbizqy	tbizqx	tborderz	tbordery	tborderx
2	1.98242188	1.68945313	1.6015625	1.62109375	1.64550781	1.98730469	0.34667969	0.33691406	0.34667969	1.62109375	1.64550781	1.98730469	1.34277344	1.71386719	1.71386719
3	1.9921875	1.70410156	1.63085938	1.62597656	1.64550781	1.99707031	0.35644531	0.33691406	0.34179688	1.62597656	1.64550781	1.99707031	1.35253906	1.72363281	1.70898438
4	2.00683594	1.70898438	1.61621094	1.63085938	1.64550781	1.99707031	0.34179688	0.3515625	0.34179688	1.63085938	1.64550781	1.99707031	1.35742188	1.72851563	1.71386719
5	1.99707031	1.70898438	1.62109375	1.63085938	1.640625	1.99707031	0.34179688	0.33691406	0.3515625	1.63085938	1.640625	1.99707031	1.35253906	1.73339844	1.71875
6	2.00683594	1.70410156	1.62597656	1.63085938	1.65039063	2.00683594	0.34667969	0.32714844	0.34179688	1.63085938	1.65039063	2.00683594	1.35253906	1.73339844	1.72363281
7	1.99707031	1.72363281	1.62109375	1.64550781	1.640625	1.98730469	0.33691406	0.32714844	0.33203125	1.64550781	1.640625	1.98730469	1.35253906	1.73828125	1.72851563
8	1.99707031	1.71386719	1.62109375	1.64550781	1.64550781	2.00195313	0.34667969	0.33691406	0.34179688	1.64550781	1.64550781	2.00195313	1.35742188	1.73828125	1.72851563
9	1.98730469	1.70410156	1.62597656	1.640625	1.65527344	2.00195313	0.34667969	0.33691406	0.34179688	1.640625	1.65527344	2.00195313	1.36230469	1.75292969	1.72851563
10	2.02148438	1.70898438	1.63085938	1.63085938	1.66992188	1.99707031	0.34179688	0.33203125	0.33691406	1.63085938	1.66992188	1.99707031	1.37207031	1.76269531	1.77246094
11	2.04589844	1.77734375	1.78222656	1.61132813	1.68945313	2.01171875	0.34667969	0.33691406	0.33203125	1.61132813	1.68945313	2.01171875	1.35253906	1.72363281	1.6796875
12	2.00195313	1.73339844	1.61132813	1.65527344	1.61621094	2.08984375	0.34667969	0.34667969	0.3515625	1.65527344	1.61621094	2.08984375	1.37207031	1.73828125	1.68457031
13	2.00683594	1.74804688	1.66503906	1.61621094	1.76269531	2.00195313	0.3515625	0.33691406	0.34179688	1.61621094	1.76269531	2.00195313	1.33789063	1.7578125	1.76269531
14	1.87011719	1.68945313	1.66015625	1.62109375	1.70410156	2.01171875	0.32714844	0.32714844	0.33691406	1.62109375	1.70410156	2.01171875	1.37695313	1.78710938	1.77246094
15	1.95800781	1.56738281	1.66015625	1.57714844	1.65527344	2.06542969	0.33691406	0.33203125	0.32714844	1.57714844	1.65527344	2.06542969	1.36230469	1.72851563	1.72851563
16	2.00195313	1.73339844	1.65039063	1.43554688	1.69433594	2.04101563	0.32714844	0.34667969	0.32714844	1.43554688	1.69433594	2.04101563	1.29882813	1.73339844	1.67480469
17	2.05566406	1.76757813	1.57714844	1.640625	1.66503906	2.01171875	0.34667969	0.32226563	0.33203125	1.640625	1.66503906	2.01171875	1.40625	1.78222656	1.78222656
18	2.08007813	1.77246094	1.87011719	1.58203125	1.69921875	2.08496094	0.33691406	0.33203125	0.33203125	1.58203125	1.69921875	2.08496094	1.38671875	1.76757813	1.70410156
19	2.00195313	1.73339844	1.61621094	1.5625	1.60644531	2.2265625	0.34179688	0.33203125	0.32714844	1.5625	1.60644531	2.2265625	1.25	1.70410156	1.66992188
20	2.09472656	1.78710938	1.69921875	1.66503906	1.68945313	2.02148438	0.32714844	0.33203125	0.32714844	1.66503906	1.68945313	2.02148438	1.41113281	1.78710938	1.76757813
21	2.02148438	1.76757813	1.81152344	1.59179688	1.70410156	2.08007813	0.34667969	0.33203125	0.34667969	1.59179688	1.70410156	2.08007813	1.35253906	1.72851563	1.69921875
22	2.00195313	1.73828125	1.62597656	1.55273438	1.65039063	2.1875	0.3515625	0.34667969	0.34179688	1.55273438	1.65039063	2.1875	1.3671875	1.74316406	1.73339844
23	2.08496094	1.73828125	1.58203125	1.61132813	1.66992188	2.02148438	0.33691406	0.33203125	0.34179688	1.61132813	1.66992188	2.02148438	1.39648438	1.79199219	1.79199219

Figura 5.9: Conjunto de datos en formato .csv

del clasificador.

Para realizar la limpieza y transformación de los datos se optó por las siguientes medidas estadísticas: media, mediana, desviación estándar, varianza, mínimo, máximo y percentile.

$$\begin{aligned} & \text{MEDIA} \\ (X) = \tilde{x} &= \frac{\sum_{i=1}^N X_i}{N} \end{aligned} \quad (5.1)$$

$$\begin{aligned} & \text{MEDIANA} \\ x_{il} + \frac{(N_M/2) - N_{i-1}}{f_i} \end{aligned} \quad (5.2)$$

$$\begin{aligned} & \text{DESVIACIONESTANDAR} \\ \sigma &= \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - x^-)^2} \end{aligned} \quad (5.3)$$

### 5.2.5. Conjunto de entrenamiento

El conjunto de entrenamiento de datos se creó una lista el cual se tendrá almacenado múltiples archivos .csv y para reconocer si el registro es de un paciente con neuropatía diabética se agregó un valor Positivo (TRUE) y para el paciente sano Falso (FALSE), el total de la lista son diez pacientes con neuropatía y cinco personas sanas y la lectura interna de las posiciones que contiene 400 registro por pacientes con 14 columnas que se encuentra almacenada otra lista.

En la evaluación de la lista de datos se trabajo con la técnica de validación cruzada (cross-validation), para evaluar los resultados de un análisis estadístico y garantizar que son independientes de la partición entre datos de entrenamiento y prueba. Existe diferentes formas de validación de datos se tomó en cuenta el más optimo para lista de datos es la validación cruzada dejando uno fuera (Leave-one-out) Figura 5.10, implica separar los datos de forma que para cada iteración tengamos una sola muestra para los datos de prueba y todo el resto conformando los datos de entrenamiento y de la misma manera en cada registro de paciente que se realizo separación por segmentos y se le asignó el tamaño a analizar.

### 5.2.6. Resultados

El rendimiento del modelo en el que se llevaron a cabo experimentos permitió observar su comportamiento al realizar la tarea de clasificación. Para aplicar los algoritmos seleccionados al modelo los experimentos fueron realizados en el lenguaje de programación *Python*, el cual se utilizó en la distribución Anaconda para adentrarse en el cómputo científico, los algoritmos estudiados: *Random Forest*, *AdabBoost*, *Maquina de soporte vectorial* (SVM). En los resultados obtenidos se observa que el algoritmo **AdaBoost** obtuvo una precisión de 87 % y por segmento un 80 % como se muestra en la Tabla 5.8, en la matriz de confusión se observa

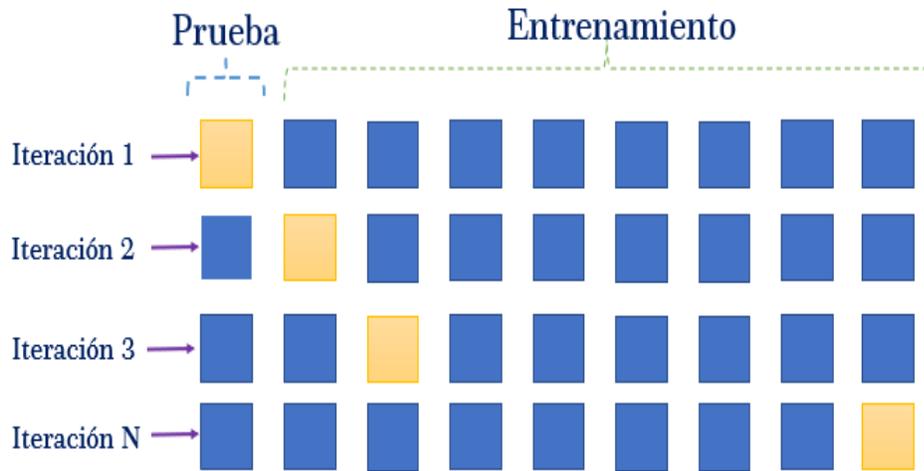


Figura 5.10: Datos de Entrenamiento y prueba

Tabla 5.7 el total de instancias clasificada correctamente fueron 9 pacientes con neuropatía diabética y 4 pacientes sanos. En la tabla 5.7 se observa por segmentos las instancias positivas clasificadas 52 y 115.

T	F	CLASS
4	1	TRUE
1	9	FALSE

Tabla 5.6: Matriz de confusión por paciente

T	F	CLASS
<b>52</b>	18	TRUE
25	<b>115</b>	FALSE

Tabla 5.7: Matriz de confusión por segmento

<b>Paciente</b>	Precisión	recall	f1-Score	Support
True	0.80	0.80	0.80	5
False	0.90	0.90	0.90	10
Avg / Total	0.87	0.87	0.87	15
<b>Segmento</b>	Precisión	recall	f1-Score	Support
True	0.67	0.73	0.70	70
False	0.86	0.82	0.84	140
Avg / Total	0.80	0.79	0.79	210

Tabla 5.8: Algoritmo AdaBoost.

<b>Paciente</b>	Precisión	recall	f1-Score	Support
True	0.75	0.60	0.67	5
False	0.82	0.90	0.86	10
Avg / Total	0.80	0.80	0.79	15
<b>Segmento</b>	Precisión	recall	f1-Score	Support
True	0.69	0.63	0.66	70
False	0.82	0.86	0.84	140
Avg / Total	0.78	0.78	0.78	210

Tabla 5.9: Algoritmo RandomForest

<b>Paciente</b>	Precisión	recall	f1-Score	Support
True	0.25	0.20	0.22	5
False	0.64	0.70	0.67	10
Avg / Total	0.51	0.53	0.52	15
<b>Segmento</b>	Precisión	recall	f1-Score	Support
True	0.39	0.37	0.38	70
False	0.69	0.71	0.70	140
Avg / Total	0.59	0.60	0.60	210

Tabla 5.10: Algoritmo SVM

# **Capítulo 6**

## **Análisis de resultados**

Las siguientes gráficas se analizará el comportamiento de cada una, y la información es obtenida por la red de sensores que es conectada: rodilla derecha, tobillo derecho, rodilla izquierda, tobillo izquierdo y espalda baja. Cada sensor de manera interna se compone de tres ejes  $X, Y$  y  $Z$ , se observa que durante la marcha de cada paciente el desplazamiento de los datos el cual se forman líneas recta o paralelas, la información esparcida o en otros casos crean ondas en el interior o exterior de las líneas de datos. En la información se observa el comportamiento de los datos que se obtiene a partir de la marcha del paciente. A continuación se describe cada una de las gráficas obtenidas de los pacientes que participaron en las pruebas.

1. La gráfica Figura 6.1, se observó que la rodilla Figura 6.1c y tobillo izquierdo Figura 6.1d, se forman curvas sobre el eje  $X$  e indica el desplazamiento de los datos, en la rodilla derecha se observó Figura 6.1a que los datos se desplaza en un solo punto específico sobre los tres ejes  $X, Y$  y  $Z$ , se forman una línea que inicializa en el punto 0.0 y termina en el punto 0.2. El tobillo derecho Figura 6.1b, en el eje  $X$  y se encuentra algunos puntos y al final de la línea se logra la formación de ondas de datos. La espalda baja Figura 6.1e en el eje  $X$  en la mayoría de los datos se encuentra más dispersos sobre la línea.

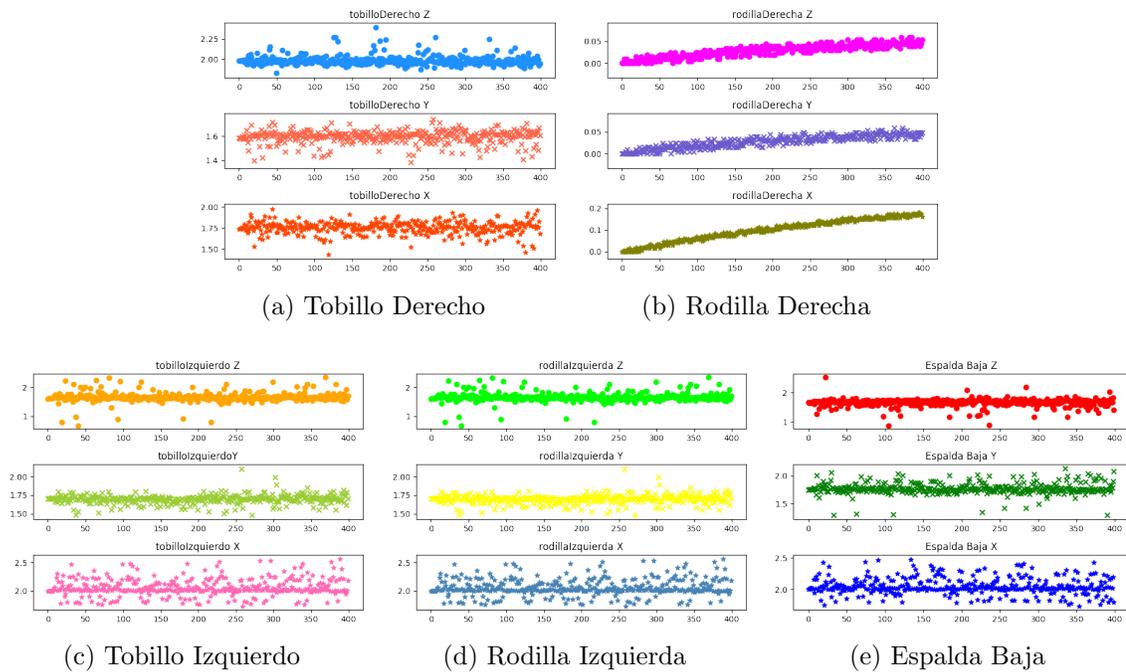


Figura 6.1: Paciente 01

2. La gráfica Figura 6.2, se observó que el tobillo Figura. 6.2d y la rodilla izquierda Figura.6.2d, en el eje  $X$ , la formación de curvas, la mayoría de los puntos se encuentra esparcidos en los extremos de la línea recta. En la rodilla derecha Figura 6.2a forma una línea paralela que inicia de abajo hacia arriba y el tobillo derecho ejes  $X$  y  $Y$  Figura 6.2a, en cambio el eje  $Z$ , los datos se encuentra centrado en un punto específico. La espalda baja 6.2e en el eje  $X$ , los datos se encuentra dispersos sobre la línea recta.

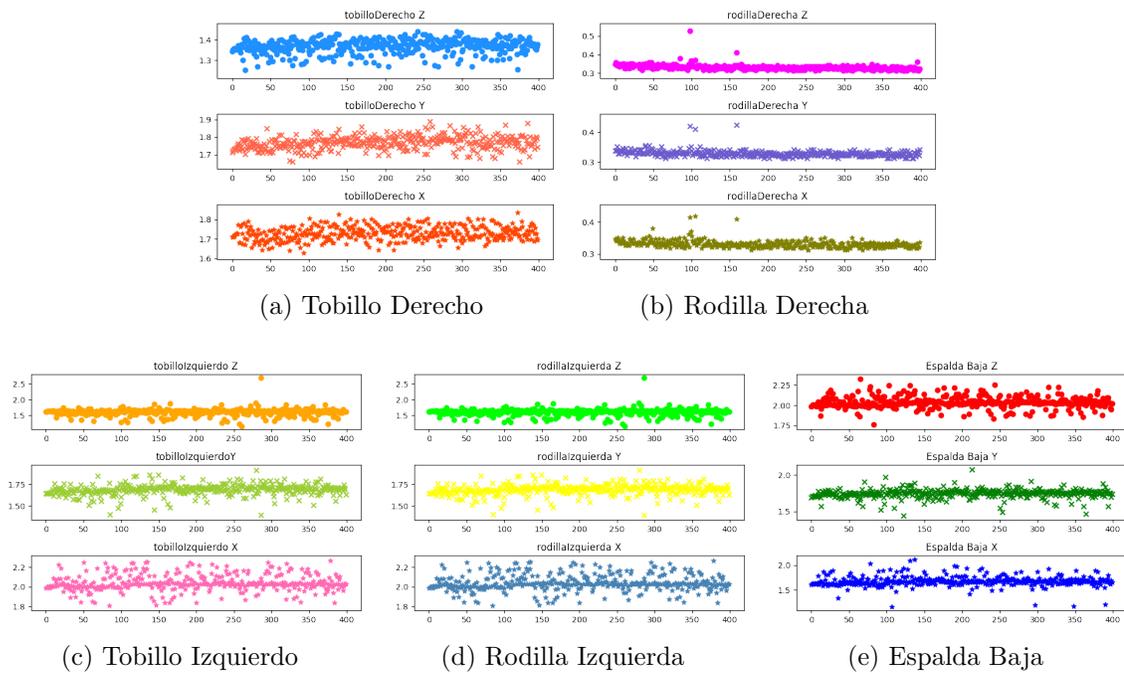


Figura 6.2: Paciente 02

3. La gráfica Figura 6.3, el tobillo izquierdo Figura 6.3d y rodilla izquierda Figura 6.3d, en el eje  $X$  se observó la formación de ondas sobre la línea de datos en algunos extremos se encuentra datos disperso. La rodilla derecha Figura 6.3a, en las tres ejes  $X, Y$ , y  $Z$  forma una línea paralela que inicializa de arriba y va decremento conforme los datos se van ingresando. El tobillo derecho Figura 6.3b, en las tres ejes  $X, Y$  y  $Z$ , se crea una línea recta y la mayoría de los datos se encuentra disperso en los extremos de la línea recta. En la espalda baja Figura 6.3e, el eje  $Z$  se forman ondas y en algunos puntos se encuentra dispersos.

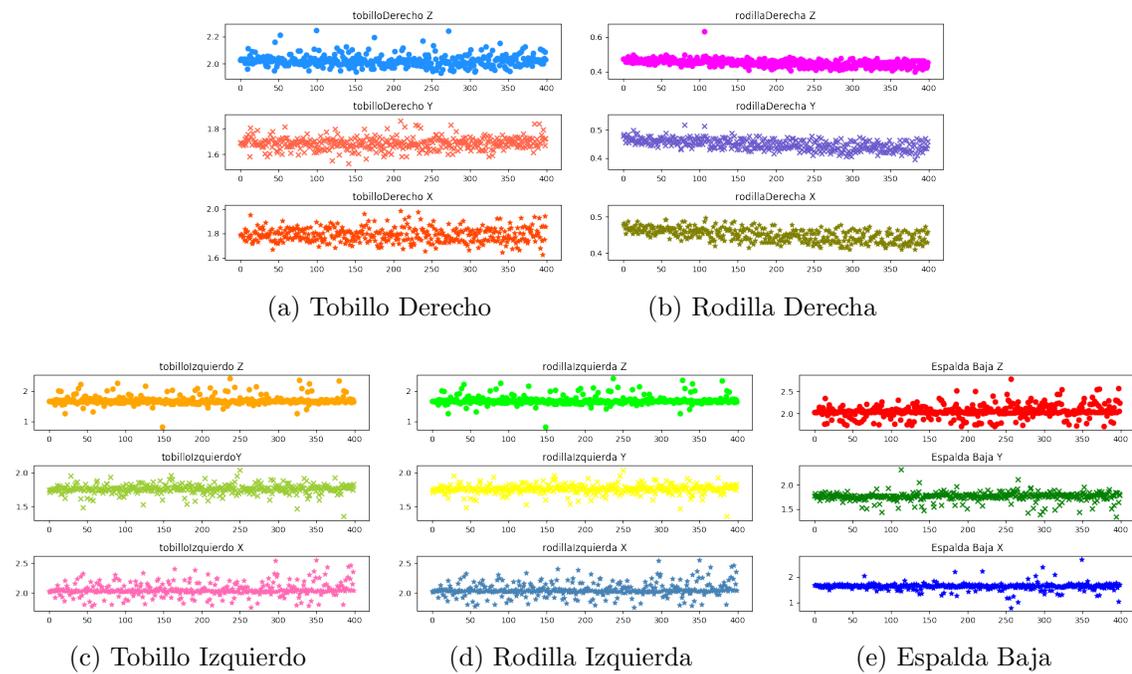


Figura 6.3: Paciente 03

4. La gráfica Figura 6.4, se observó que los ejes  $X, Y$  y  $Z$  de los cinco sensores, al inicio los datos se separan de la línea y después todos los datos forman una línea recta de datos.

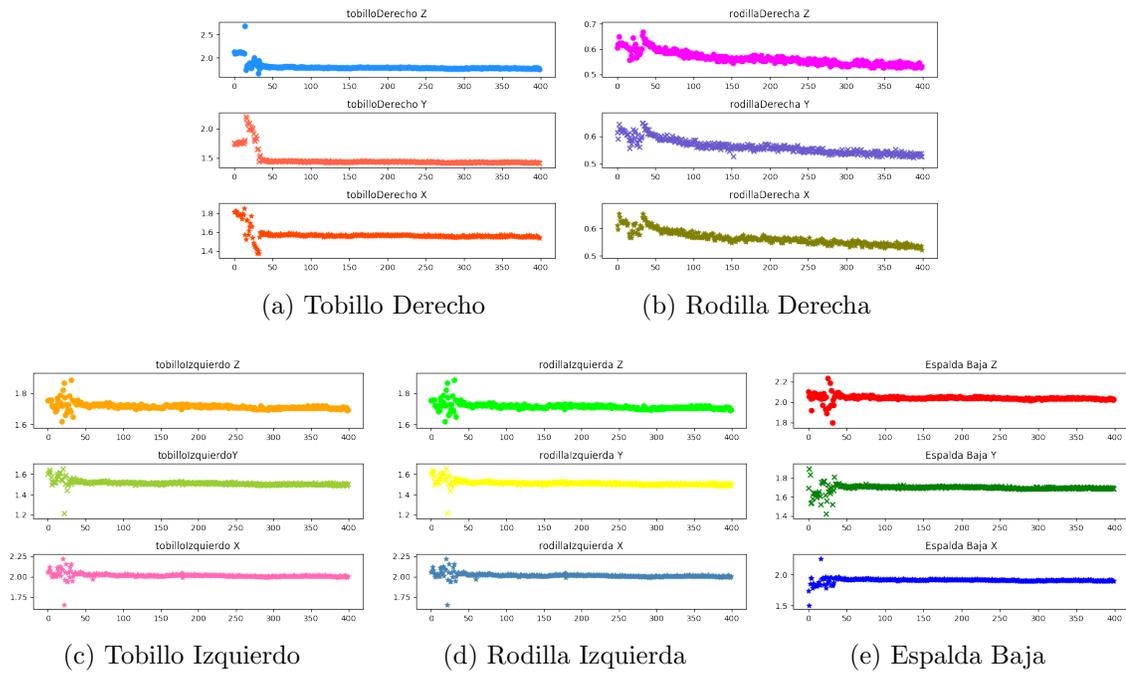


Figura 6.4: Paciente 04

5. La gráfica Figura 6.5, se observó el eje  $X$  del tobillo izquierdo Figura 6.5d y la rodilla izquierda Figura 6.5d, se forman curvas en el eje  $X$  y los datos se encuentra esparcidos. El tobillo derecho Figura 6.5b en la eje  $Y$ , se forman curvas sobre la línea datos, en el eje  $X$  se forman ondas en su mayoría los datos se encuentra dispersos, la rodilla derecha Figura 6.5a, se observó que línea recta, en el eje  $X$  y sus datos se encuentra mas dispersos en comparación de los ejes  $Y$  y  $Z$ . La espalda baja 6.5e, en el eje  $Z$  se forma ondas sobre la línea recta de datos.

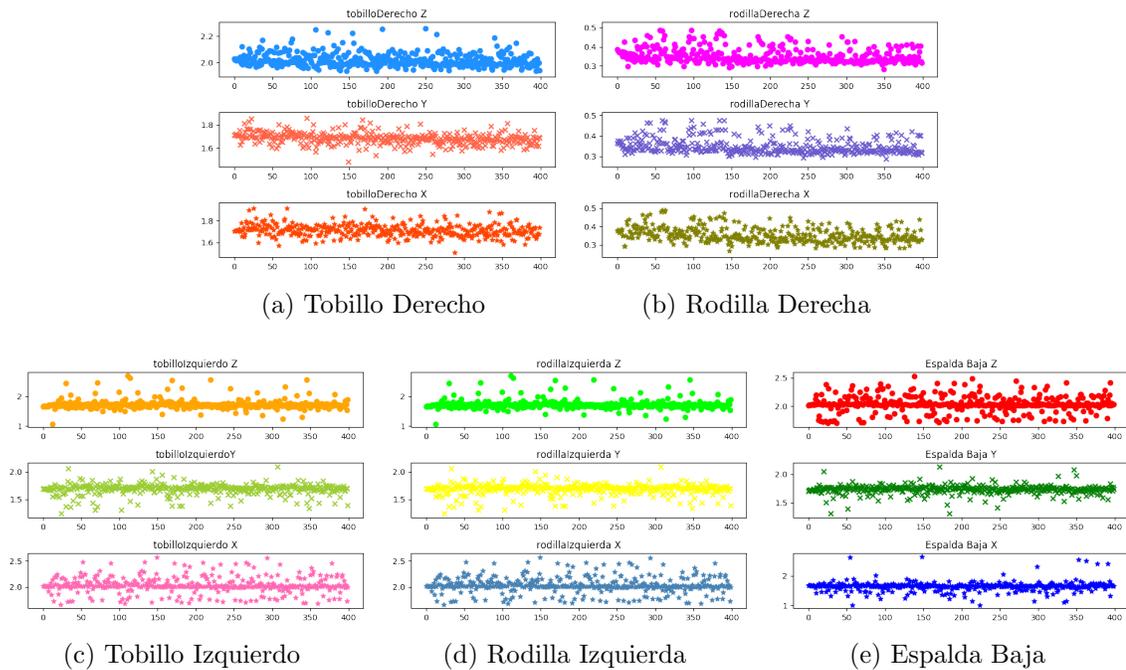


Figura 6.5: Paciente 05

6. La gráfica Figura 6.6, se observó que eje  $X$  del tobillo izquierdo Figura 6.6d y rodilla izquierda Figura 6.6c, los datos se encuentra esparcidos sobre la línea de datos. El tobillo derecho Figura 6.6b el eje  $X$  los datos se encuentra dispersos, en comparación a los ejes  $Y$  y  $Z$ , los datos se encuentra dispersos sobre la línea de datos, la rodilla derecha Figura 6.6a, se observó en las tres ejes  $X$ ,  $Y$  y  $Z$ , los datos se encuentra esparcido sobre la línea de datos. La espalda baja Figurafig:esp05, el eje  $Z$ , se observó que los datos se encuentra arriba de la línea de datos y al mismo tiempo se encuentra unidos.

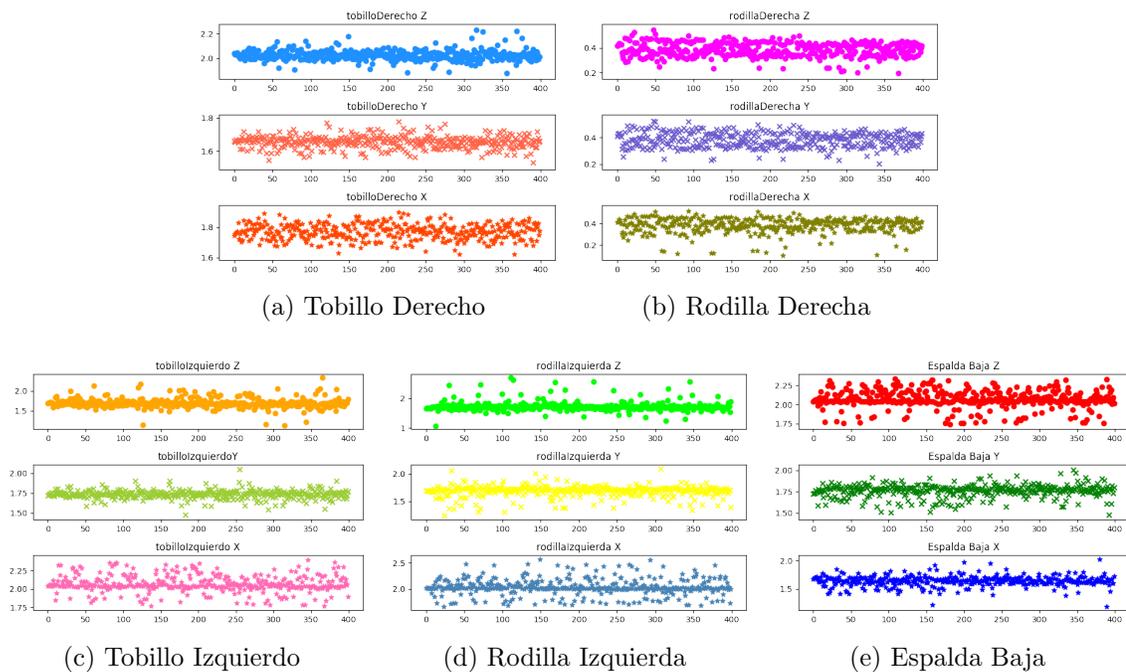


Figura 6.6: Paciente 06

7. La gráfica Figura 6.7, se observó que el tobillo izquierdo Figura 6.7d y la rodilla izquierda Figura 6.6c en el eje  $X$ , los datos se encuentra dispersos sobre la línea recta. El tobillo derecho Figura 6.7b, el eje  $X$ , se forman curvas en la línea recta los ejes  $Y$  y  $Z$ , que forma una línea recta y los puntos se encuentra encimados. La rodilla derecha en las tres posiciones  $X, Y$  y  $Z$ , se crea una línea recta de datos, se encuentra dispersos los datos bajo de línea recta. La espalda baja en la posición  $Z$ , se crean sobre la línea de datos.

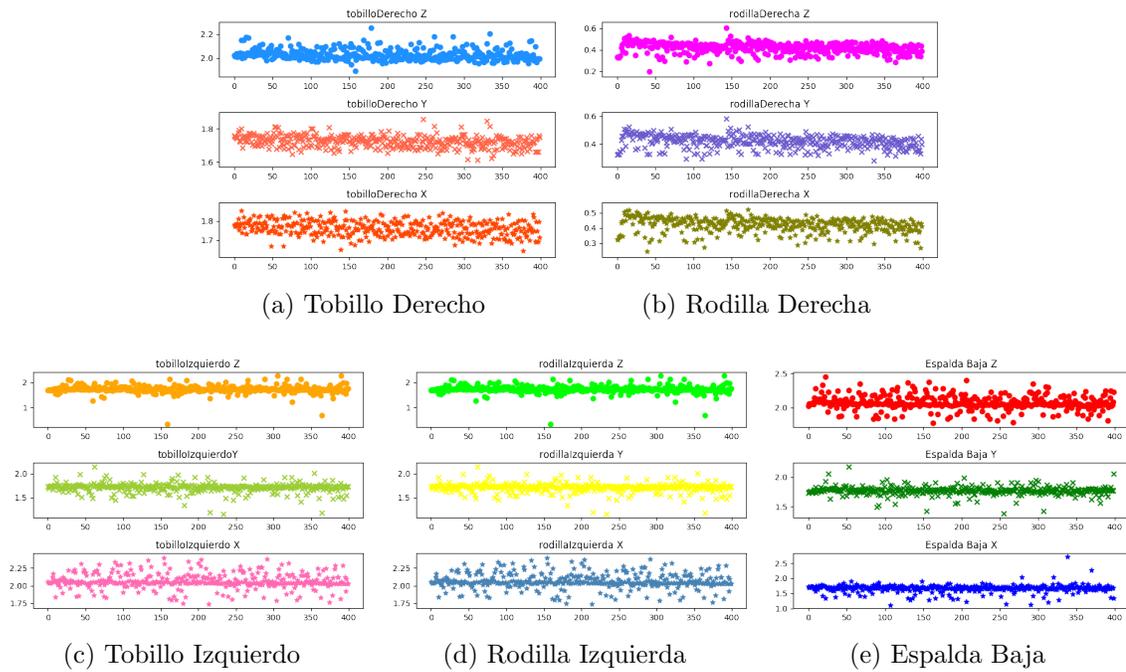


Figura 6.7: Paciente 07

8. La gráfica Figura 6.8, se observó que el tobillo izquierdo Figura 6.8d y la rodilla izquierda Figura 6.8c, el eje  $X$  los datos se encuentra dispersos de la línea recta en comparación de a los ejes  $Y$  y  $Z$ . El tobillo derecho Figura 6.8b, el eje  $Y$ , se forma una línea en curva en el interior de la línea de datos y en las tres ejes  $X, Y$  y  $Z$ . los datos se encuentra dispersos. La rodilla derecha Figura 6.8a en los tres ejes  $X, Y$  y  $Z$ , se creo una línea vertical de datos. La espalda baja Figura 6.8e, el eje  $Z$ , la mayor parte datos se encuentra dispersos a los lados de línea recta.

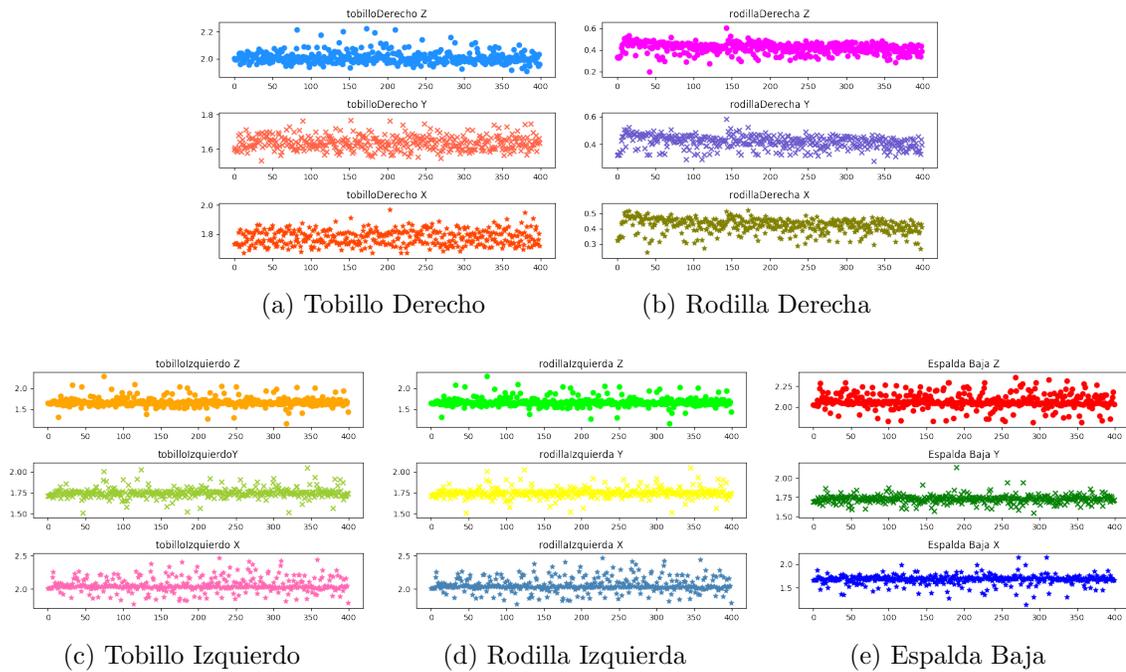


Figura 6.8: Paciente 08

9. La gráfica 6.9, el tobillo izquierdo y rodilla izquierda se crea ondas de datos que pasa sobre la línea recta y algunos datos se encuentra dispersos sobre la línea. El tobillo derecho en las tres posiciones  $X$ ,  $Y$  y  $Z$ , los datos se encuentra dispersos entres y se logra la formación de una línea recta. La rodilla derecha se crea una línea recta datos que va descendiendo conforme a la obtención de información. La espalda baja en la posición  $Z$ , los datos se encuentra dispersos sobre la línea de datos.

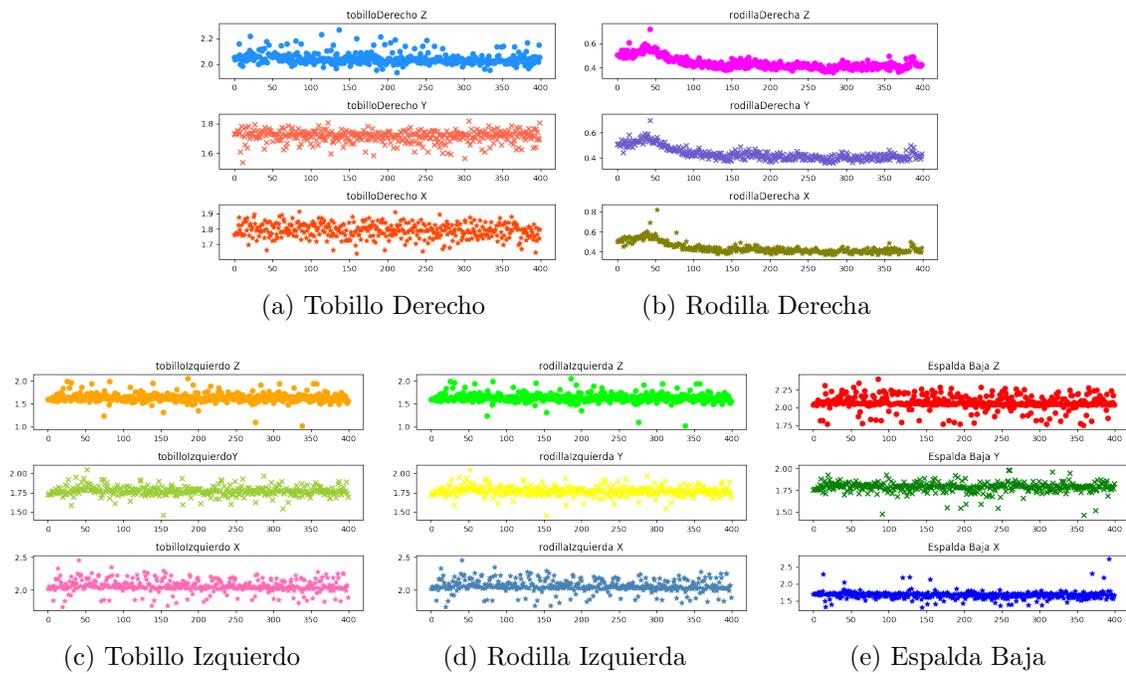


Figura 6.9: Paciente 09

10. La gráfica Figura 6.10, el tobillo izquierdo Figura 6.10d y rodilla izquierda Figura 6.10c, el eje  $X$  los datos se encuentra esparcidos en los extremos de la línea de datos. El tobillo derecho Figura 6.10b, se observó en los ejes  $X$  y  $Y$ , los datos se encuentra dispersos y se forma una línea datos. La rodilla derecha Figura 6.10a, en los ejes  $X$ ,  $Y$  y  $Z$  forma una línea recta datos. La espalda baja Figura 6.10e, el eje  $Z$ , en algunos puntos los datos se encuentra dispersos sobre la línea de datos.

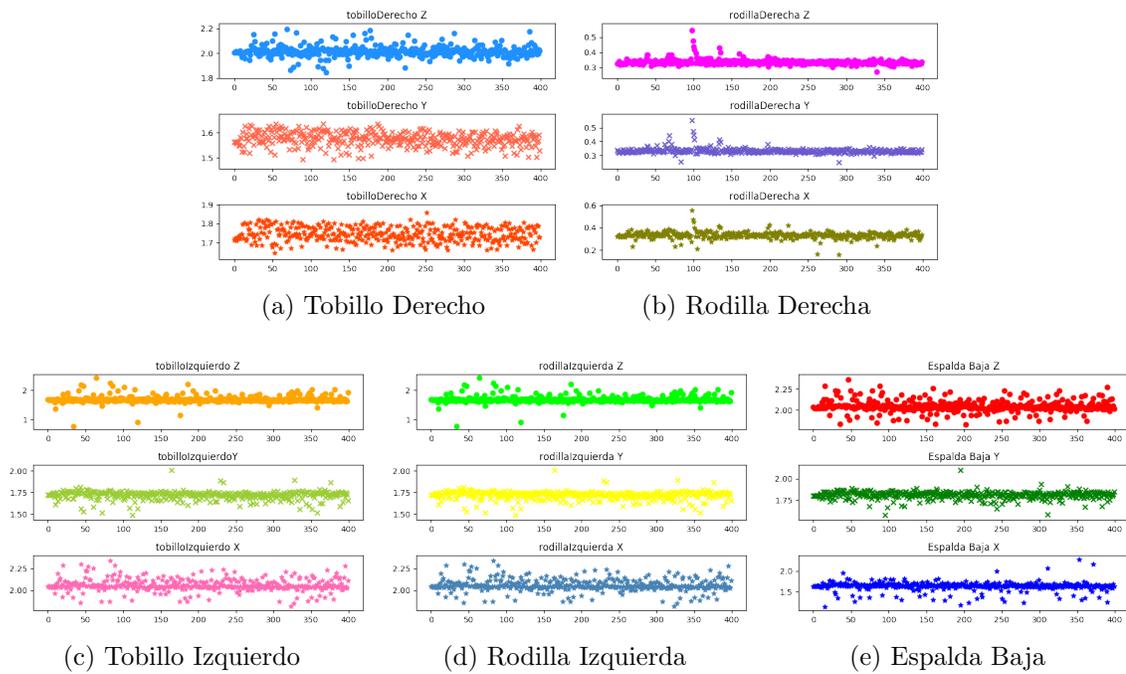


Figura 6.10: Paciente 10

11. La gráfica Figura 6.11, el tobillo izquierdo 6.11d y rodilla izquierda Figura 6.11c, los datos del eje  $X$  se encuentra dispersos sobre la línea recta de datos. El tobillo derecho los ejes  $X$  y  $Y$ , los datos se encuentra sobre la línea recta. La rodilla derecha Figura 6.11a, en los ejes  $X$ ,  $Y$  y  $Z$  una línea recta datos en la que muestra se encuentra dispersos en los extremos de la línea. La espalda baja 6.11e en el eje  $Z$ , se crea una línea recta y en algunas puntos se encuentra disperso los datos.

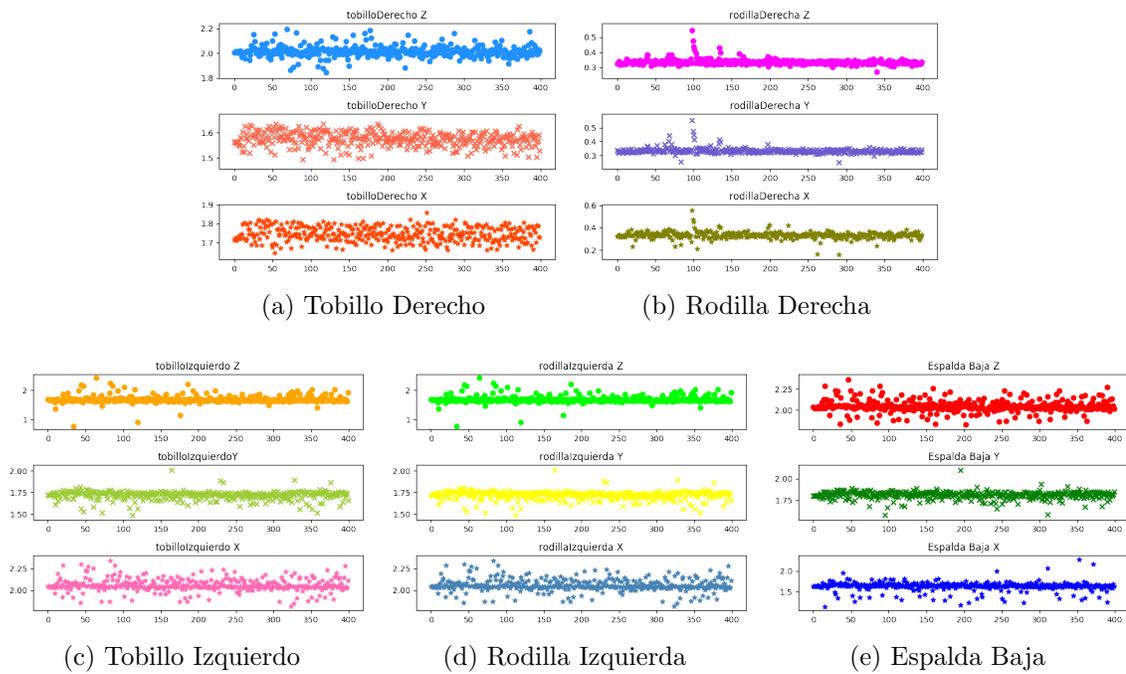


Figura 6.11: Paciente 11

12. La gráfica Figura 6.12, el tobillo izquierdo Figura 6.12d y rodilla derecha Figura 6.11c, el eje  $Z$  se forman ondas en el interior de la línea de datos. El tobillo derecho Figura 6.11b en los ejes  $X$  y  $Y$ , se crean ondas en el interior de la línea de datos. La rodilla derecha Figura 6.12a, los ejes  $X$ ,  $Y$  y  $X$  se forma una línea paralela de datos. La espalda baja Figura 6.12e, en el eje  $Z$  los datos se encuentra dispersos sobre la línea de datos.

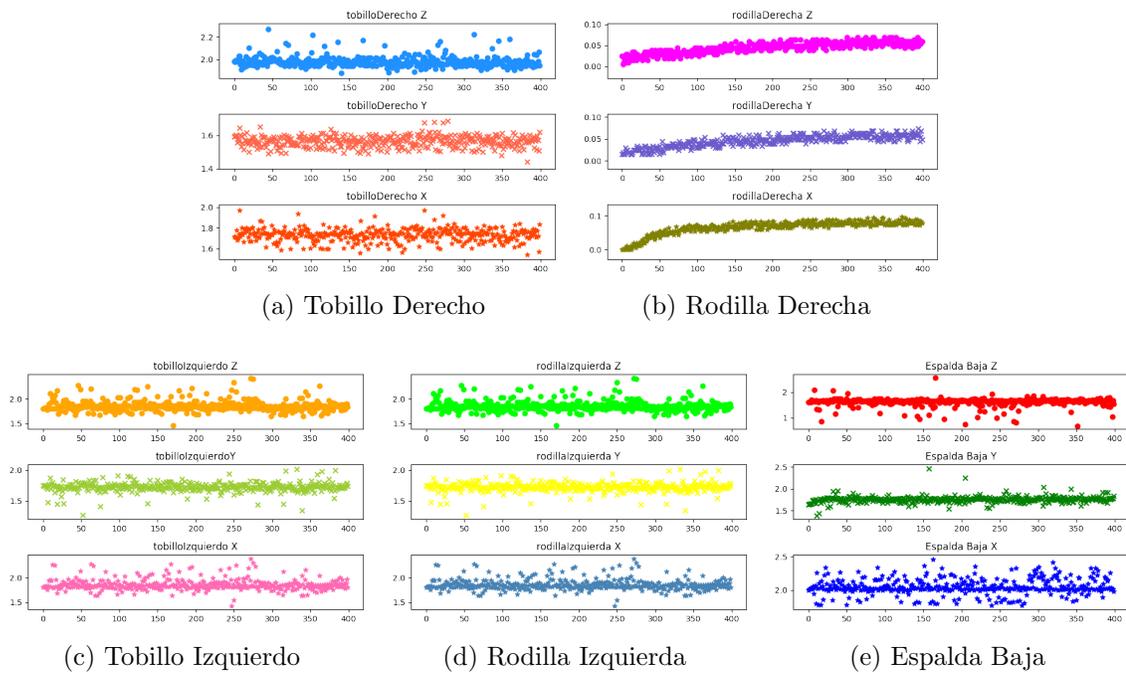


Figura 6.12: Paciente 12

13. La gráfica Figura 6.13, el tobillo izquierdo Figura 6.13d y rodilla izquierda Figura 6.13c en el eje  $X$ , los datos se encuentra dispersos en los extremos de la línea recta. La rodilla derecha Figura 6.13c en los ejes  $X$  y  $Y$  se crea ondas en el interior de la línea recta. El tobillo derecho Figura 6.13b se crea una línea paralela que en los tres ejes  $X$ ,  $Y$  y  $Z$ . La espalda baja Figura 6.13e en el eje  $X$ , se crean ondas de datos sobre la línea.

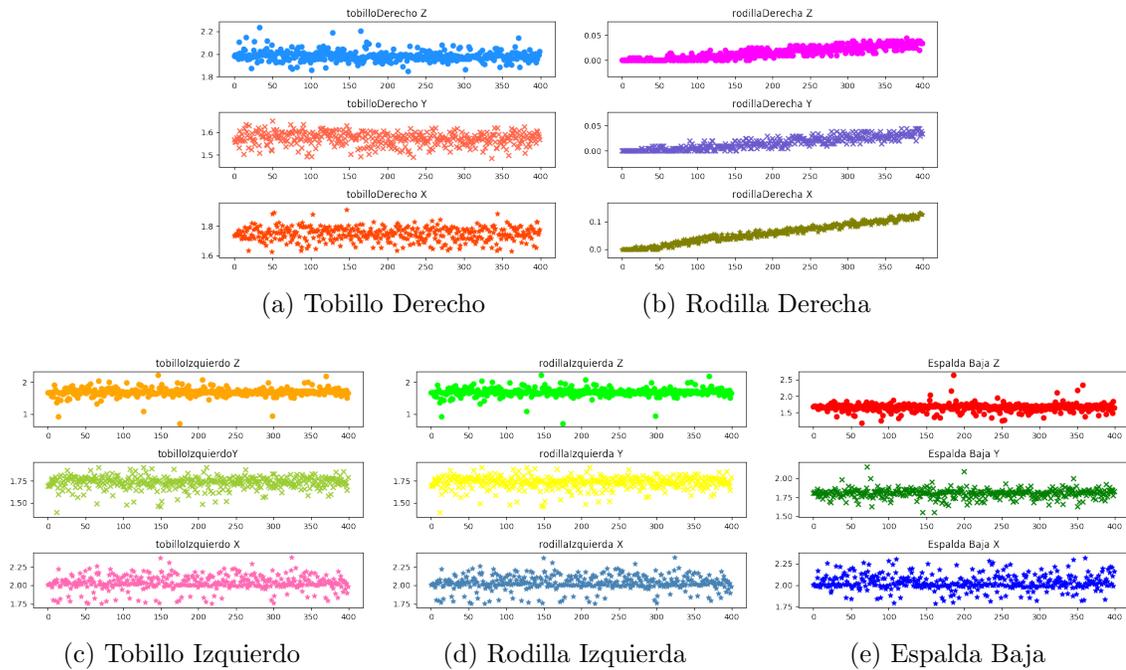


Figura 6.13: Paciente 13

14. La gráfica 6.14, el tobillo izquierdo Figura 6.14d y rodilla izquierda Figura 6.14c en el eje  $X$ , se forman ondas en el interior de la línea de datos. El tobillo derecho Figura 6.14b en los ejes  $X$  y  $Y$  se crea ondas y la información se encuentra dispersos en los extremos de la línea. El tobillo derecho 6.14b en los ejes  $X$ ,  $Y$  y  $Z$ , los datos se encuentra muy dispersos alrededor y no se logra apreciar la formación de línea de datos. La espalda baja Figura 6.14e en la posición  $Z$ , la información se encuentra dispersos sobre la línea.

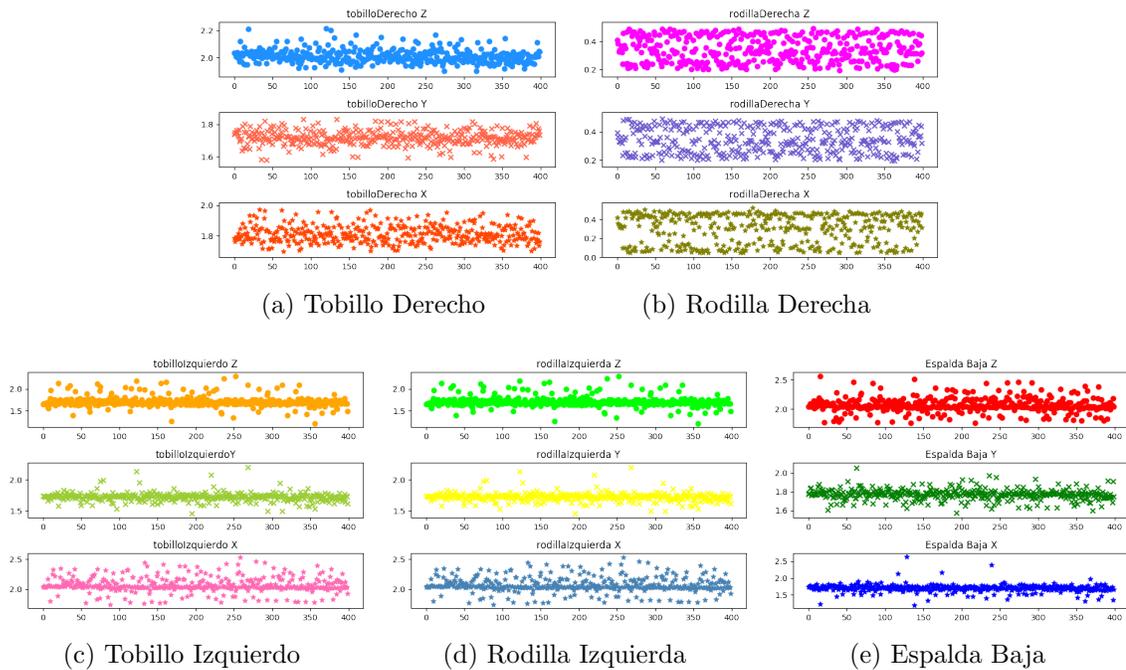


Figura 6.14: Paciente 14

15. La gráfica Figura 6.15, la rodilla izquierda Figura 6.15c en el eje  $X$  se crean ondas sobre la línea recta de datos. El tobillo izquierdo Figura 6.15d, los ejes  $X$  y  $Y$  se forman ondas en el interior de la línea recta de datos. La rodilla derecha Figura 6.15a en los tres ejes  $X$ ,  $Y$  y  $Z$ , se observó que se forma paralela que empieza de arriba hacia abajo y la información se esparce conforme la línea se va en decremento. El tobillo derecho Figura 6.15b los datos se encuentra en el eje  $Y$  y se encuentra dispersos en los extremos de la línea recta. La espalda baja Figura 6.15e, en el eje  $Z$ , se encuentra dispersa los datos sobre la línea tratando de crear ondas en los extremos de la línea recta.

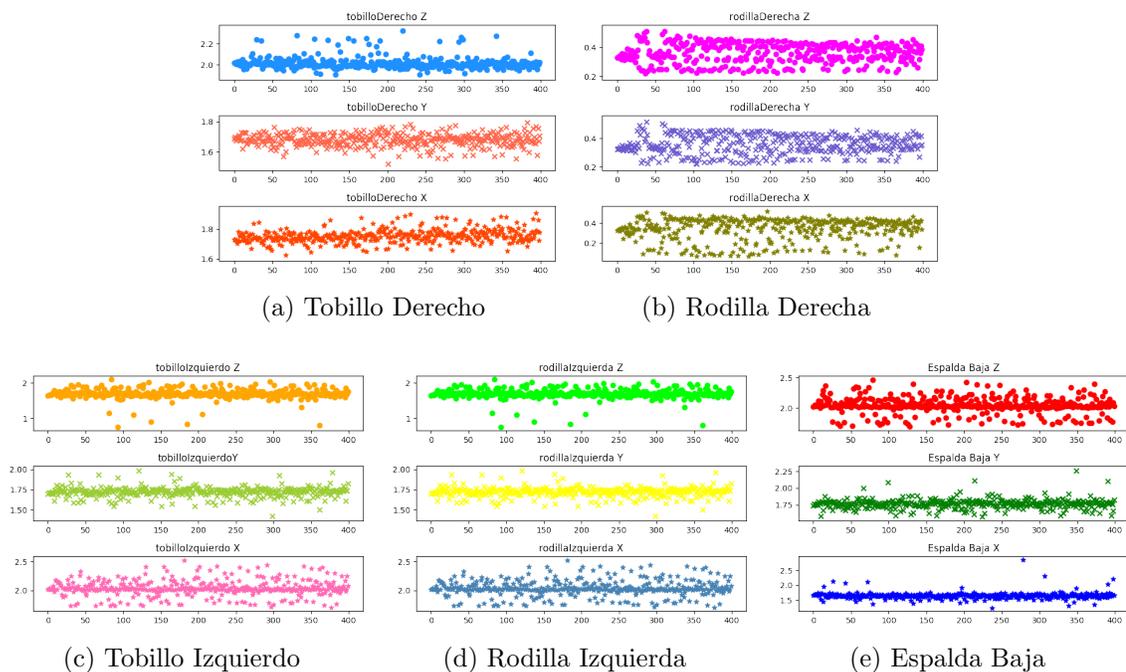


Figura 6.15: Paciente 15

El análisis se concluye los diferentes cambios que presenta cada uno de los tres ejes que contiene el sensor. El sensor conectado en la rodilla derecha se observó que al obtener la información los datos crean una línea recta o paralela a sí mismo lo mismo para el tobillo derecho se llegó a la conclusión hubo un defecto de fabricación, en comparación a los sensores que fueron conectados en los ejes  $Y$  y  $Z$  de la rodilla izquierda y tobillo izquierdo, y cada registro se observa que son similares unos a los otros en comparación al sensor conectado en el eje  $X$  y se tienen cambios significativos que surgen durante la caminata del paciente. Toda la información que fue extraída se obtiene durante el movimiento de cada uno de los pacientes.

## **Capítulo 7**

# **Conclusiones y Trabajos Futuros**

## 7.1. Conclusiones

La presente tesis se tuvo como objetivo diseñar y construir una red de sensor y desarrollar un clasificador de neuropatía diabética utilizando diferentes algoritmos de aprendizaje automático.

La construcción de la red sensores se trabajo construyeron una red que solo obtuviera información en una sola pierna y saber el tiempo en que se recoleta los datos. Después obteniendo los resultado del primer experimento se realizó la construcción de una red sensores que se conecto en las dos piernas la cual se obtiene mas información para realizar análisis.

Durante la captura de los datos se observó que los niños se adaptaron más fácil a la red de sensores y al realizar la caminata la información se obtenía se almacena mas rápida, en comparación en los adultos que algunos casos se sentía nervioso al realizar la caminata y el tiempo de información fue mas largo.

En el preprocesamiento se utilizaron nueve medidas estadísticas y durante los experimentos con los diferentes algoritmos se observo que los resultados que se obtuvieron eran entre 70 al 50 %, y se opto utilizar solo tres medidas estadísticas media, mediana y desviación estándar las cuales se obtuvieron óptimos resultados algoritmos estudiados.

## 7.2. Trabajos Futuros

Si bien la presente tesis abordó como el tema de neuropatía diabética el cual afecta a la comunidad diabética en México por la obesidad, mal control de la presión arterial y glucosa, el tratamiento farmacéutico inadecuado y falta de herramientas tecnológicas para la clasificación de neuropatía diabética. Para dar seguimiento a la investigación temas interesantes a tratar puede ser:

- Alimentar continuamente la base de datos de personas con neuropatía diabética y sanas
- Desarrollar una nueva red de sensores con un modelo diferente sensor.
- Desarrollar un sistema experto en donde se ingrese el registro y predecir si el paciente se encuentra sano o enfermo.
- Realizar pruebas con diferentes algoritmos de aprendizaje automático o con redes neuronales.
- Crear una red de sensores de manera inalámbrica.
- Crear un DataSet con personas que sufran esclerosis.
- Crear una red sensores en donde sus conexiones se encuentre conectados a partir de las manos, codos para personas con osteoartritis.

# Referencias

- Bernardino, F. M., Sánchez-DelaCruz, E., y Ruíz, I. V. M. (2018). Knee-ankle sensor for gait characterization: Gender identification case. En *International symposium on intelligent computing systems* (pp. 31–40).
- Bourgeois, A. B., Mariani, B., Aminian, K., Zambelli, P., y Newman, C. (2014). Spatio-temporal gait analysis in children with cerebral palsy using, foot-worn inertial sensors. , *39*(1), 436 - 442.
- Breiman, L. (2001). Random forests. , *45*(1), 5–32.
- Calhoun, M., Longworth, M., y Chester, V. L. (2011). Gait patterns in children with autism. , *26*(2), 200 - 206.
- Clark, C. C., Barnes, C. M., Holton, M., Summers, H. D., y Stratton, G. (2016). Profiling movement quality and gait characteristics according to body mass index in children 9 y 11 years. , *49*, 291 - 300.
- Díaz, A. O. (2014). Algoritmo multclasificador con aprendizaje incremental que manipula cambios de conceptos.
- Ferreira, M. á. r. A., Artur J y Figueiredo. (2012). Algoritmos de impulso: una revisión de métodos, teoría y aplicaciones. , 35–85.
- Flores, A., Maldonado, S., y Weber, R. (2015). Selección de atributos y support vector machines adaptado al problema de fuga de clientes.
- Garcia, E., y Lozano, F. (2007, 01). Boosting support vector machines. , 153-167.
- Jamil, N., Khir, N. H. M., Ismail, M., y Razak, F. H. A. (2015). Gait-based emotion detection of children with autism spectrum disorders: A preliminary investigation. , *76*, 342 - 348.
- Kavakiotis, I., Tsave, O., Salifoglou, A., Maglaveras, N., Vlahavas, I., y Chouvarda, I. (2017). Machine learning and data mining methods in diabetes research. , *15*, 104 - 116.
- Mannini, A., Martinez-Manzanera, O., Lawerman, T. F., Trojaniello, D., Della Croce, U., Sival, D. A., . . . Sabatini, A. M. (2017). Automatic classification of gait in children with early-onset ataxia or developmental coordination disorder and controls using inertial sensors. , *52*, 287–292.
- Molina, R., Rodríguez, C., y cols. (2012). Definición, clasificación y diagnóstico de la diabetes mellitus. , *10*, 7–12.
- Otero, J., y Sánchez, L. (2006). Induction of descriptive fuzzy classifiers with the logitboost algorithm. , *10*(9), 825–835.

- Ramírez-López, P., Giles, O. A., y Avilés, A. G. P. (2017). Neuropatía diabética: frecuencia, factores de riesgo y calidad de vida en pacientes de una clínica de primer nivel de atención. , *19*(4), 105–111.
- Raschka, S. (2015). *Python machine learning*. Packt Publishing Ltd.
- Rosenbaum, D., Westhues, M., y Bosch, K. (2013). Effect of gait speed changes on foot loading characteristics in children. , *38*(4), 1058 - 1060.
- Shulcloper, J., y Vega, S. (2012). Combinación de clasificadores supervisados: estado del arte. (4122).
- Vadim, K. (2018). Overview of different approaches to solving problems of data mining. , *123*, 234 - 239.
- Witten, E., Ian H y Frank. (2005). *Data mining: herramientas y técnicas prácticas de aprendizaje automático (series morgan kaufmann en sistemas de gestión de datos)* (Vol. 104).
- Yu, S., Chen, H., Wang, Q., Shen, L., y Huang, Y. (2017). Invariant feature extraction for gait recognition using only one uniform model. , *239*, 81 - 93.

## Anexos

El siguiente código fue elaborado con la herramienta MATLAB para extraer la información durante la marcha del paciente en tiempo real.

```
1
2 clear all
3 clc
4
5
6 a = arduino();
7
8 for i=1:400;
9     % tobillo derecho:
10    tobilloDerechoZ(i) = readVoltage(a,12);
11    tobilloDerechoY(i) = readVoltage(a,13);
12    tobilloDerechoX(i) = readVoltage(a,14);
13    % tobillo izquierdo:
14    tobilloIzquierdoZ(i) = readVoltage(a,9);
15    tobilloIzquierdoY(i) = readVoltage(a,10);
16
17    tobilloIzquierdoX(i) = readVoltage(a,11);
18    % rodilladerecha:,re
19    rodillaDerechaZ(i) = readVoltage(a,6);
20    rodillaDerechaY(i) = readVoltage(a,7);
21    rodillaDerechaX(i) = readVoltage(a,8);
22    % rodilla izquierda:
23    rodillaIzquierdaZ(i) = readVoltage(a,3);
24    rodillaIzquierdaY(i) = readVoltage(a,4);
25    rodillaIzquierdaX(i) = readVoltage(a,5);
26    % columna:
27    columnaZ(i) = readVoltage(a,0);
28    columnaY(i) = readVoltage(a,1);
29    columnaX(i) = readVoltage(a,2);
30
31    x = [tobilloDerechoZ(i), tobilloDerechoY(i), tobilloDerechoZ(i),
rodillaIzquierdaZ(i), rodillaIzquierdaY(i), rodillaIzquierdaX(i),
tobilloDerechoZ(i), tobilloDerechoY(i), tobilloDerechoX(i),
tobilloIzquierdoZ(i), tobilloIzquierdoY(i),tobilloIzquierdoX(i),columnaZ(i)
), columnaY(i), columnaX(i)];
32    disp(x);
33
34 end
35
36
37    tdz = vec2mat(tobilloDerechoZ,1);
38    tdy = vec2mat(tobilloDerechoY,1);
39    tdx = vec2mat(tobilloDerechoX,1);
40    tiz = vec2mat(tobilloIzquierdoZ,1);
41    tiy = vec2mat(tobilloIzquierdoY,1);
42    tix = vec2mat(tobilloIzquierdoX,1);
```

```

43 rdz = vec2mat(rodillaDerechaZ,1);
44 rdy = vec2mat(rodillaDerechaY,1);
45 rdx = vec2mat(rodillaDerechaX,1);
46 riz = vec2mat(rodillaIzquierdaZ,1);
47 riy = vec2mat(rodillaIzquierdaZ,1);
48 rix = vec2mat(rodillaIzquierdaZ,1);
49 cz = vec2mat(columnaZ,1);
50 cy = vec2mat(columnaY,1);
51 cx = vec2mat(columnaX,1);
52
53 f = [tdz tdy tdx tiz tiy tix rdz rdy rdx tiz tiy tix cz cy cx];
54
55 xlswrite('patronesCaminar.xlsx', f, 1, 'B2:P500');

```

### Índice de algoritmos 7.1: Código de obtención de datos de la red de sensores

A continuación se muestra el código construido en PYTHON y en el se realiza los experimentos con los diferentes algoritmos.

```

1
2 import os
3 import glob
4 import numpy as np
5 import pandas as pd
6 import random
7 from scipy import stats
8 from sklearn.ensemble import RandomForestClassifier
9 from sklearn.datasets import make_classification
10 from sklearn.ensemble import AdaBoostClassifier
11 from sklearn import svm
12 #'colz','coly','colx','rdizqz','rdizqy','rdizqx','rderz','rdery','rderx','
    tbizqz','tbizqy','tbizqx','tbdery','tbdery','tobderx'
13 #'colz','coly','colx','rdizqz','rdizqy','rdizqx','tbizqz','tbizqy','tbizqx','
    tbdery','tbdery','tobderx'
14 #'colz','coly','colx','tbizqz','tbizqy','tbizqx','tbdery','tbdery','tobderx'
15 #'colz','coly','colx','rdizqz','rdizqy','tbizqx','rderz','rdery','rderx'
16 #'rdizqz','rdizqy','rdizqx','rderz','rdery','rderx'
17 #'tbizqz','tbizqy','tbizqx','tbdery','tbdery','tobderx'
18 #'colz','coly','colx','rdizqz','rdizqy','rdizqx','tbizqz','tbizqy','tbizqx'
19 #'colz','coly','colx','rderz','rdery','rderx','tbdery','tbdery','tobderx'
20 #'rdizqz','rdizqy','rdizqz','tbizqx','tbizqy','tbizqx'
21 #'rderz','rdery','rderx','tbdery','tbdery','tobderx'
22
23
24 random.seed(1330)
25
26 path = 'C:\\\\Users\\fabiola\\Documents\\pruebasNeuropatia\\Pacientes\\Pacientes
    con neuropatia\\' #Ruta de archivo
27 def get_merged_csv(path, fmask_, header=0):
28     fmask, clase=fmask_
29     return [(clase,pd.read_csv(f, index_col=None, usecols=['colz','coly','colx
    ','rdizqz','rdizqy','rdizqx','rderz','rdery','rderx','tbizqz','tbizqy','

```

```

30     tbizqx', 'tbderz', 'tbdery', 'tobderx']))
31         for f in glob.glob(os.path.join(path, fmask))][0]
32
33 Y_REAL=[]
34 Y_PRED=[]
35 Y_REAL_ACC=[]
36 Y_PRED_ACC=[]
37 T=True
38 F=False
39
40 # df_list contiene los .csv con la ruta en donde se encuentra guardado cada
41 # una y la creacion de un mascara de cada .csv
42 df_list = [get_merged_csv(path, fmask)
43             for fmask in [('*paciente1.csv',T),(*paciente2.csv',T),(*
44 #for fmask in [('*paciente6.csv',T),(*paciente7.csv',T),(*
45 #for fmask in [('*paciente1.csv',T),(*paciente2.csv',T),(*
46 ]
47
48 step=30 #se determina el numero de datos para realizar el entrenamiento por
49 # los segmentos separados
50 width=30
51 for i in range(0,len(df_list)): # iteracion de los indices de la lista de
52 # elementos
53     primero=df_list.pop(0) #elimina el primer dato en la iteracion de la lista
54     df_seg=[] #contiene la lista de la iteracion
55     y_train=[] #entrenamiento con los datos de la lista
56     for clase,df_list_ in df_list: # iteracion de la clase contiene el
57     # nombre de cada columna con registros de cada paciente
58     # for i in range(0,len(df_list_[0:400]),step): # iteracion en la lista
59     # en donde se encuentra almacenado el numero de datos
60     # df_seg.append(df_list_[i:i+width]) # append () agrega un obj
61     # pasado en la lista existente.
62     # y_train.append(clase)
63
64     X_train=[] #
65     for seg in df_seg: # se
66         uno= seg.std().as_matrix()

```

```

64     dos= seg.mean().as_matrix()
65     tres= seg.median().as_matrix()
66     cuatro = seg.max().as_matrix()
67     cinco= seg.min().as_matrix()
68     seis=seg.var().as_matrix()
69     siete= np.percentile(seg.as_matrix(),25,axis=0)
70     ocho= np.percentile(seg.as_matrix(),75,axis=0)
71
72
73     #print(tres)
74
75     X_train.append(np.hstack([uno,dos,tres]))
76
77
78
79     print(len(X_train))
80     print(len(y_train))
81
82
83     from sklearn.preprocessing import StandardScaler
84     # pruebas con RandomForest
85
86     scaler=StandardScaler()
87     X_train=scaler.fit_transform(X_train)
88
89     clf= AdaBoostClassifier(n_estimators=100)
90     #clf = svm.SVC(kernel="linear",C=2)
91     #clf = RandomForestClassifier(50)
92     #print(y_train)
93
94     clf.fit(X_train, y_train)
95     #print(clf.feature_importances_)
96
97
98     if X_train[-1].shape==150:
99         X_train.pop()
100        y_train.pop()
101
102        df_seg=[]
103        y=[]
104        for clase,df_list_ in [primero]:
105            for i in range(0,len(df_list_[0:400]),step):
106                df_seg.append(df_list_[i:i+width])
107                y.append(clase)
108                Y_REAL_ACC.append(clase)
109                X=[]
110                for seg in df_seg:
111                    uno= seg.std().as_matrix()
112                    dos = seg.mean().as_matrix()
113                    tres = seg.median().as_matrix()
114                    cuatro = seg.max().as_matrix()

```

```

115         cinco= seg.min().as_matrix()
116         seis = seg.var().as_matrix()
117         siete= np.percentile(seg.as_matrix(),25,axis=0)
118         ocho= np.percentile(seg.as_matrix(),75,axis=0)
119
120
121         #X.append(np.hstack([uno, dos, tres, seis, siete, ocho]))
122         X.append(np.hstack([uno, dos, tres]))
123         #X.append(seg.as_matrix().flatten())
124
125     Y_REAL.append(clase)
126
127
128     df_list.append(primero)
129     X=scaler.transform(X)
130     pre=clf.predict(X) #predecir la clasificacion de X
131     Y_PRED_ACC+=[x for x in pre]
132     total_true=np.sum(pre)
133     print(pre)
134     if total_true/len(pre)>=0.5:
135         Y_PRED.append(True)
136     else:
137         Y_PRED.append(False)
138
139
140 print(Y_REAL)
141 print(Y_PRED)

```

Índice de algoritmos 7.2: Código en PYTHON para la clasificación de algoritmos