



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

**TECNOLÓGICO NACIONAL
DE MÉXICO**

**INSTITUTO TECNOLÓGICO
SUPERIOR DE HUATUSCO**

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

OPCIÓN: TESIS

TRABAJO PROFESIONAL

**“DISEÑO DE UN TRAJE DE RED DE SENSORES DE INERCIA
CORPORAL PARA REGISTRAR, ANALIZAR Y RECONOCER
ACTIVIDADES RELACIONADAS CON EL CULTIVO DE CAFÉ”**

**QUE PARA OBTENER EL GRADO DE:
MAESTRO EN INGENIERÍA**

PRESENTA:

DAGOBERTO MAGÍN MURILLO

DIRECTOR DE TESIS:

LUIS HUMBERTO SÁNCHEZ MEDEL

CODIRECTOR DE TESIS:

MIGUEL ÁNGEL SOLÍS JIMÉNEZ



HUATUSCO, VERACRUZ, MÉXICO

AGOSTO, 2023



OFICIO DE APROBACIÓN DE LA TESIS

**"DISEÑO DE UN TRAJE DE RED DE SENSORES DE INERCIA CORPORAL PARA REGISTRAR,
ANALIZAR Y RECONOCER ACTIVIDADES RELACIONADAS CON EL CULTIVO DE CAFÉ"**

Tesis

Que como parte de los requisitos para obtener el título de:
Maestro en Ingeniería

Presenta:
Ing. Dagoberto Magín Murillo

Comité Tutorial

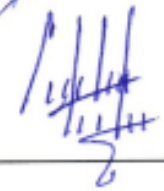
M. C. Luis Humberto Sánchez Medel
Presidente



M.I. Miguel Ángel Solís Jiménez
Secretario



Dr. César Leonardo Melchor
Hernández
Vocal



M.I. Rafael Tejeda García
Vocal



Huatusco de Chicuéllar, Ver.

Agosto 2023

DEDICATORIA

A mis familiares, amigos, maestros y todos aquellos quienes me demostraron su gran apoyo incondicional, me han aportado sus conocimientos, guiado y me han ofrecido las mejores enseñanzas de la vida, siendo mi apoyo en todo momento.

RECONOCIMIENTOS

A mis Asesores:

Agradezco a mis asesores que gracias a sus conocimientos, consejos y correcciones hicieron posible la elaboración de este proyecto.

A mi familia:

Por la confianza depositada en mí, por todos los sacrificios y esfuerzos realizados para que pueda lograr terminar con mis estudios, ¡Gracias!

RESUMEN

En este documento se describe el desarrollo de una red de sensores corporal. Las redes de sensores corporales *Body Sensor Network* (BSN) forman una tecnología que permite la adquisición de datos y en la actualidad están generando gran interés debido a las aplicaciones en diferentes ámbitos tanto científicos como industriales para la realización de investigaciones y control de procesos. La mayoría de los desórdenes músculo esqueléticos son problemas que se presentan entre los agricultores debido a los riesgos ergonómicos y las condiciones de trabajo asociadas durante la recolección de café.

Es por ello que se plantea el diseño de una arquitectura, basada en redes de sensores corporales, que permita funcionar como herramienta en la detección de las actividades de recolección de café la cual realiza un análisis en tiempo real durante el proceso de esta actividad debido a que los trabajadores están sujetos a movimientos físicos repetitivos que pueden conducir a diferentes riesgos ergonómicos.

Posteriormente se hace uso de aprendizaje automático *Machine Learning* para presentar los modelos matemáticos con la intención de justificar las decisiones tomadas de la actividad de recolección de café. Por último, se describe los resultados, obtenidos llegando a las conclusiones planteadas de esta manera obtiene el diseño de un traje de red de sensores de inercia corporal electrónico para registrar las diferentes actividades estableciendo un nivel de seguridad en la recolección de café. Como aportación de este trabajo se tiene la vinculación de la BSN con las nuevas tecnologías que son los dispositivos vestibles para generar la base de conocimiento.

ABSTRACT

This document describes the development of a body sensor network. The Body Sensor Network (BSN) is a technology that allows the acquisition of data and is currently generating great interest due to its applications in different fields, both scientific and industrial, for research and process control. Most musculoskeletal disorders are problems among farmers due to ergonomic risks and associated working conditions during coffee harvesting.

That is why the design of an architecture is proposed, based on networks of body sensors, which allows it to function as a tool in the detection of coffee harvesting activities, which performs an analysis in real time during the process of this activity due to that workers are subjected to repetitive physical movements that can lead to different ergonomic risks.

Subsequently, machine learning is used to present the mathematical models with the intention of justifying the decisions made in the coffee harvesting activity. Finally, the results are described, obtained reaching the conclusions raised in this way, the design of an electronic body inertia sensor network suit is obtained to record the different activities, establishing a level of security in the coffee harvest. As a contribution of this work, there is the link between the BSN and the new technologies that are wearable devices to generate the knowledge base.

Contenido

GLOSARIO	9
INTRODUCCIÓN.....	10
PLANTEAMIENTO DEL PROBLEMA	11
JUSTIFICACIÓN.....	12
HIPÓTESIS	13
OBJETIVOS	14
OBJETIVO GENERAL	14
OBJETIVOS ESPECÍFICOS.....	14
CAPÍTULO 1. ANTECEDENTES	15
1.1. DISPOSITIVOS VESTIBLES.....	15
1.1.1. Características.....	17
1.1.2. Tipos de wearables	18
1.2. FUNDAMENTOS DE MACHINE LEARNING	19
1.2.1. Aprendizaje supervisado.....	19
1.2.2. Aprendizaje no supervisado.....	20
1.2.3. Aprendizaje reforzado	21
1.3. Clasificación de datos	22
1.4. Red de sensores corporales	29
1.4.1. BSN	29
1.4.2. WBSN.....	32
1.5. Características estadísticas.....	33
CAPÍTULO 2. ESTADO DEL ARTE	38
CAPÍTULO 3. APLICACIÓN DE LA METODOLOGÍA	44
3.1. Introducción	44

3.3.	Análisis de sensores	45
3.3.1.	Acelerómetros.....	46
3.3.2.	Giroscopios	47
3.4.	OpenLog Artemis.....	49
3.5.	Diseño	50
3.5.1.	Posicionamiento de los sensores respecto a la persona	51
3.6.	Adquisición de datos.....	52
3.7.	Preprocesamiento	56
3.8.	Extracción de características.....	59
3.9.	Creación del modelo.	62
CAPÍTULO 4. RESULTADOS Y DISCUSIÓN		68
4.1.	Modulo Chest.....	68
4.2.	Modulo Left Arm.....	80
4.3.	Modulo RA	93
5.	CONCLUSIONES	106
REFERENCIAS		111
APÉNDICE		114
Características del Giroscopio		114
Características del Acelerómetro		115
Esquemático sensor OpenLog Artemis.....		116
Dimensiones de Sensor OpenLog Artemis		117
Base para batería del Wearable Device Modelo 3D		118
Base para sensor del Wearable Device Modelo 3D.....		119
Cubierta superior del Wearable Device Impresión 3D		120
Módulo Chest Impresión 3D.....		121

Módulo Left Arm Impresión 3D.....	122
Módulo Right Arm Impresión 3D.....	123
ANEXOS	124
Código para Aplicación en Android	124
Script para obtener un conjunto de modelos de ML.	130

LISTA DE FIGURAS

FIGURA 1 Pronóstico de crecimiento del mercado wearable	16
FIGURA 2 Proceso de Machine Learning	19
FIGURA 3 Aprendizaje supervisado.....	20
FIGURA 4 Aprendizaje no supervisado.....	21
FIGURA 5 Aprendizaje reforzado	21
FIGURA 6 Logistic Regression	23
FIGURA 7 Logistic Regression	24
FIGURA 8 Decision Tree Classifier.....	24
FIGURA 9 Support Vector Machine.....	25
FIGURA 10 Linear Regression	26
FIGURA 11 Elección del modelo	27
FIGURA 12 Arquitectura de un BSN.....	30
FIGURA 13 Arquitectura de WBSN.....	32
FIGURA 14 Metodología a seguir	48
FIGURA 15 OpenLog Artemis	49
FIGURA 16 Batería LiPo a 3.7 V	49
FIGURA 17 Modelo en 3D base para la batería	50
FIGURA 18 Modelo 3D para sensor OLA.....	51
FIGURA 19 Modelo 3D cubierta superior	51
FIGURA 20 Rotación de ejes.....	52
FIGURA 21 Ubicación de Dispositivos Vestibles	52
FIGURA 22 Proceso de recolección de café	53
FIGURA 23 Módulos implementados en el usuario	53
FIGURA 24 Etiquetado de actividades	54
FIGURA 25 Adquisición de datos.....	55
FIGURA 26 Archivo CSV	56
FIGURA 27 Preprocesamiento de datos	57
FIGURA 28 Ventana temporal deslizante.....	58
FIGURA 29 Señal preprocesada	59

FIGURA 30 Características Estadísticas.....	61
FIGURA 31 Señal con características estadísticas.....	62
FIGURA 32 Classification Learner.....	63
FIGURA 33 Decision Trees.....	63
FIGURA 34 Workspace.....	63
FIGURA 35 Summary.....	64
FIGURA 36 Prediction Model.....	65
FIGURA 37 Confusion matrix.....	65
FIGURA 38 Validation (ROC) Curve.....	66
FIGURA 39 Coarse Tree.....	66
FIGURA 40 Medium Tree.....	67
FIGURA 41 Fine Tree.....	67
FIGURA 42 Summary Coarse Tree CH.....	68
FIGURA 43 Prediction Model Coarse Tree CH.....	69
FIGURA 44 Confusion Matrix Coarse Tree CH.....	70
FIGURA 45 ROC Coarse Tree CH.....	71
FIGURA 46 Coarse Tree CH.....	72
FIGURA 47 Summary Medium Tree CH.....	73
FIGURA 48 Prediction Model Medium Tree CH.....	73
FIGURA 49 Matrix Confusion Medium Tree CH.....	74
FIGURA 50 ROC Medium Tree CH.....	75
FIGURA 51 Medium Tree CH.....	76
FIGURA 52 Summary Fine Tree CH.....	77
FIGURA 53 Prediction Model Fine Tree CH.....	77
FIGURA 54 Matrix Confusion Fine Tree CH.....	78
FIGURA 55 ROC Fine Tree CH.....	79
FIGURA 56 Fine Tree CH.....	80
FIGURA 57 Summary Coarse Tree LA.....	81
FIGURA 58 Prediction Model Coarse Tree LA.....	81
FIGURA 59 Matrix Confusion Coarse Tree LA.....	82
FIGURA 60 ROC Coarse Tree LA.....	83

FIGURA 61 Corase Tree LA.....	84
FIGURA 62 Summary Medium Tree LA.....	85
FIGURA 63 Prediction Model Medium Tree LA	85
FIGURA 64 Matrix Confusion Medium Tree LA.....	86
FIGURA 65 ROC Medium Tree LA.....	87
FIGURA 66 Medium Tree LA	88
FIGURA 67 Summary Fine Tree LA	89
FIGURA 68 Prediction Model Fine Tree LA.....	89
FIGURA 69 Matrix Confusion Fine Tree LA	90
FIGURA 70 ROC Fine Tree LA	91
FIGURA 71 Fine Tree LA.....	92
FIGURA 72 Summary Coarse Tree RA.....	93
FIGURA 73 Prediction Model Coarse Tree RA	94
FIGURA 74Matrix Confusion Coarse Tree RA.....	95
FIGURA 75 ROC Coarse Tree RA	96
FIGURA 76 Coarse Tree RA	97
FIGURA 77 Summary Medium Tree RA	97
FIGURA 78 Prediction Model Medium Tree RA	98
FIGURA 79 Matrix Confusion Medium Tree RA	99
FIGURA 80 ROC Medium Tree RA.....	100
FIGURA 81 Medium Tree RA.....	101
FIGURA 82 Summary Fine Tree RA.....	102
FIGURA 83 Prediction Model Fine Tree RA	103
FIGURA 84 Matrix Confusion Fine Tree	103
FIGURA 85 ROC Fine Tree RA	104
FIGURA 86 Fine Tree RA	105
FIGURA 87 Resultados Módulo Chest	107
FIGURA 88 Resultados Módulo Left Arm	108
FIGURA 89 Resultados Módulo Right Arm.....	110

LISTA DE TABLAS

Tabla 1 Sensores de uso común en BSN (Lai et al., 2013)	31
Tabla 2 Clasificación de los tipos de aplicación de los dispositivos wearable.....	41
Tabla 3 Análisis de movimientos	45
Tabla 4 Características seleccionadas.....	60

LISTA DE ECUACIONES

ECUACIÓN 1 Logistic Regression.....	22
ECUACIÓN 2 Linear Regression	26
ECUACIÓN 3 Función Mean	34
ECUACIÓN 4 Función Harmmean.....	34
ECUACIÓN 5 Función Kurtosis	35
ECUACIÓN 6 Función RMS	35
ECUACIÓN 7 Función Peak2peak	36
ECUACIÓN 8 Función Skewness.....	36
ECUACIÓN 9 Función Trimmean.....	37
ECUACIÓN 10 Varianza	37
ECUACIÓN 11 Desviación Estándar.....	37

GLOSARIO

BSN: Body Sensor Network.

CH: Chest

CSV: Comma Separated Values

EMG: Electromyography

FNR: False Negative Rate

IA: Artificial Intelligence

IMU: Inertial Measurement Unit

HAR: Human Activity Recognition

LA: Left Arm

ML: Machine Learning

OLA: OpenLog Artemis

RA: Right Arm

ROC: Receiver Operating Characteristic

TPR: True Positive Rate

WD: Wearable Device

WT: Wearable Technology

WBSN: Wireless Body Sensor Network

INTRODUCCIÓN

Desde la evolución de los seres humanos han estado creando muchos tipos de herramientas para realizar diversas actividades de una manera más sencilla. Esto ha generado la creatividad del ser humano para la invención de diferentes máquinas que faciliten la vida humana al cumplir con diversas necesidades de la vida, así como también diferentes áreas como la salud, seguridad, deporte, informática, entre otras. El aprendizaje automático (ML) y las redes de sensores corporales (BSN) son ejemplos de ello.

Una red de sensores corporales *Body Sensor Network* (BSN) consiste en un sistema independiente que se utiliza para monitorear las actividades de la vida diaria de una persona. Consiste en nodos de sensores inteligentes que no obstaculizan las actividades de la vida diaria y son útiles para detectar enfermedades crónicas como ataques cardíacos, asma, diabetes, etc. y para advertir a los pacientes en caso de condiciones de emergencia. Las BSN están formados por sensores de parámetros fisiológicos colocados en el cuerpo humano, en la superficie del cuerpo o alrededor del cuerpo. Su objetivo principal es proporcionar una plataforma integrada de tecnología de comunicación alámbrica, hardware y software de computación, y una condición esencial para el desarrollo futuro de sistemas de monitoreo de atención médica mediante sensores inerciales *Inertial Measurement Unit* (IMU) por sus siglas en inglés. Los sensores inerciales son aquellos dispositivos que obtienen el reconocimiento automático de las actividades físicas que realizan los humanos, también es conocido como (HAR), *Human Activity Recognition*. Estos dispositivos pueden utilizarse para la toma de decisiones inteligente. Los sensores inerciales pueden interactuar con algunos otros dispositivos como los wearables devices que se están convirtiendo cada vez más en parte importante de la tecnología y su uso está pasando de ser simples accesorios a aplicaciones más especializadas y prácticas gracias al aprendizaje automático (ML) que se utiliza para enseñar a las máquinas cómo manejar los datos de manera más eficiente, su objetivo es aprender de los datos sin ser programadas explícitamente.

PLANTEAMIENTO DEL PROBLEMA

De acuerdo con la Organización Mundial De Salud aproximadamente 1710 millones de personas tienen trastornos musculoesqueléticos en todo el mundo(*Trastornos Musculoesqueléticos*, n.d.). Siendo as el dolor lumbar es el más frecuente, con una prevalencia de 568 millones de personas. Los trastornos musculoesqueléticos son la principal causa de discapacidad en todo el mundo, y el dolor lumbar es la causa más frecuente y el principal factor que contribuye a la carga general de trastornos musculoesqueléticos de discapacidad en 160 países. Estos padecimientos limitan en gran cantidad la movilidad y la destreza, lo que provoca una salida prematura, es decir, ausentismo laboral o pérdida de productividad, jubilaciones anticipadas, menores niveles de bienestar y una menor capacidad de participación social. (*Trastornos Musculoesqueléticos*, n.d.).

Las actividades agrícolas son unas de las practicas que más actividades físicas comprende entre tareas y posturas de realizar con una alta probabilidad de sufrir lesiones musculares por las posturas forzadas o cargas pesadas, es importante mejorar las condiciones de la recolección de café agregando y acondicionando la comodidad así como la ergonomía en las posiciones adoptadas durante el proceso de recolección del café, los movimientos repetidos realizados, el tiempo y además la cantidad del peso que llevan consigo mismos. Esto ha generado un impacto en problema de salud pública por la magnitud que pueden alcanzar este tipo de trastornos en la población agrícola trabajadora debido a la importancia que representa el cultivo del café en el panorama económico, por lo que es necesario buscar y proporcionar mejoras para los riesgos ergonómicos y sus desórdenes musculoesqueléticos asociados en la producción agrícola. Con la ayuda de una red de sensores corporales podemos registrar las actividades, movimientos y señales del cuerpo humano. La demanda de estos dispositivos aumenta día con día y varios aspectos como confiabilidad, calidad de servicio van mejorando.

JUSTIFICACIÓN

El uso de *Machine Learning* en la actualidad es importante ya que permite automatizar tareas complejas y tomar decisiones basadas en datos a gran escala. Con el creciente volumen de las bases de datos generados por la mayoría de las empresas y las organizaciones, el *Machine Learning* se ha convertido en una herramienta esencial para extraer información valiosa de esos datos. Ante esta situación que se plantea se propone el desarrollo de un dispositivo que sea capaz de registrar las actividades en las personas que se desempeñan en la recolección del café, para con la intención de realizar mejoras durante el proceso de recolección de café, para ello se ha seleccionado el uso que los sensores inerciales ya que son aptos para evaluar el movimiento humano que puede realizar una persona al llevar a cabo una actividad física gracias a las características eléctricas, mecánicas y de medición que se obtienen de ellos, de esta manera se creara la base de conocimiento para posteriormente extraer sus características procesamiento los datos y predecir los resultados. Es así como el uso de sensores inerciales en conjunto con *Machine Learning* en aplicaciones médicas es un campo que va en crecimiento porque permite tomar decisiones más precisas y basadas en datos registrados de las personas ya que su objetivo es aprender de los datos y llegar a resultados o conclusiones y es quien aportara las soluciones al problema a partir de la información proporcionada por la toma de datos de los sensores lo que puede conducir a una mayor eficiencia y una mejor calidad.

HIPÓTESIS

La utilización de sensores corporales de inercia utilizando técnicas relacionadas con aprendizaje maquina mediante modelos de clasificación multivariantes permitirán detectar la posición de los usuarios que realizan en la actividad del corte del café.

OBJETIVOS

OBJETIVO GENERAL

Diseñar un traje de red de sensores corporales de inercia para registrar las posiciones y actividades físicas durante el proceso de recolección de café, para conformar una base de conocimientos y un modelo de *Machine Learning*.

OBJETIVOS ESPECÍFICOS

- Identificar los dispositivos que permiten detectar las actividades físicas de las personas.
- Seleccionar los sensores adecuados para la detección del movimiento.
- Diseñar un dispositivo de traje de sensor corporal para llevar a cabo el registro de las actividades físicas.
- Adquirir los datos de las actividades realizadas para procesar la información
- Crear la base de conocimientos con la adquisición de datos obtenida.
- Crear un modelo basado en *Machine Learning* para reconocer las actividades humanas a partir de la base de conocimiento.

CAPÍTULO 1. ANTECEDENTES

1.1. DISPOSITIVOS VESTIBLES

Los términos *wearables*, dispositivos portátiles o también conocida como tecnología portátil se refieren a pequeños dispositivos electrónicos y móviles, o computadoras con capacidad de comunicaciones inalámbricas que se incorporan en dispositivos, accesorios o ropa, que se pueden usar en el cuerpo humano, o incluso versiones invasivas como microchips o tatuajes inteligentes.(Luczak et al., n.d.) Gracias a la comunicación inalámbrica, la eficiencia en las tecnologías de visualización avanzadas se han combinado para crear dispositivos inteligentes de última generación y con esto se espera que mejoren las partes tecnológicas y socioculturales de nuestras vidas.

La definición del término “*wearable*” proviene del anglosajón que significa literalmente “que se puede llevar puesto”. Así pues, la tecnología *wearable* es aquella electrónica diseñada para ser vestida, bien como complemento o bien como parte de algún material usado en la ropa.(Carlos López & Guerra, 2016)

Los *Wearable technology devices* o bien los podemos llamar dispositivos portátiles comprende a todos los productos tecnológicos electrónicos o dispositivos informáticos que están diseñados para llevarse cómodamente en el cuerpo. *Wearable technology* (WT) tiende a proporcionar capacidad sensorial y de escaneo, como la retroalimentación biológica y el seguimiento de la función fisiológica. (Wearable Technology, Patent Landscape Analysis, n.d.) Los WT pueden desempeñar muchas de las tareas de los ordenadores y los dispositivos móviles también tienen capacidad de comunicación que permite a los humanos acceder a los datos en tiempo real utilizando dispositivo que se encuentren conectados, son más sofisticados ya que incorporan sensores para interactuar con el cuerpo y con el entorno.

Los WT están destinados a su aplicación en dispositivos que mejoran constantemente el bienestar, además de proporcionar comodidad e informática a los usuarios.

El uso de dispositivos portátiles inteligentes puede aportar eficiencia y optimización a las aplicaciones, mejorar la calidad de vida y aumentar la productividad, así como la seguridad. Los avances en las redes móviles de baja potencia, la disminución del tamaño de los dispositivos y

sensores electrónicos, así como las ventajas que pueden proporcionar estos dispositivos han permitido el desarrollo de la WT a un ritmo rápido. El uso de los dispositivos vestibles se puede ver en el crecimiento del mercado, así como los numerosos beneficios y nuevas aplicaciones desde la perspectiva y necesidades de los usuarios. Los principales usos son la ayuda en la atención médica, el estado físico, el envejecimiento, las discapacidades, la educación, el transporte, la empresa, las finanzas, los sistemas de entrada, los juegos, la música y muchos otros. Utilizar un dispositivo vestible puede pronosticar alguna enfermedad mediante un seguimiento continuo de la salud e incluso informar al médico de manera automática para tomar medidas y prevenir enfermedades, los rastreadores de actividad simples ya son capaces de monitorear los patrones de sueño, la frecuencia cardíaca, el nivel de estrés o la temperatura corporal que podrían utilizarse para mejorar los hábitos de salud de cualquier persona. (Kresser C., n.d.) De acuerdo con la investigación de IDTechEx afirmó que el mercado de la tecnología portátil superará los 30.000 millones de dólares en 2016 y, a partir de entonces, seguirá creciendo 10 años después logra ver en la FIGURA 1.

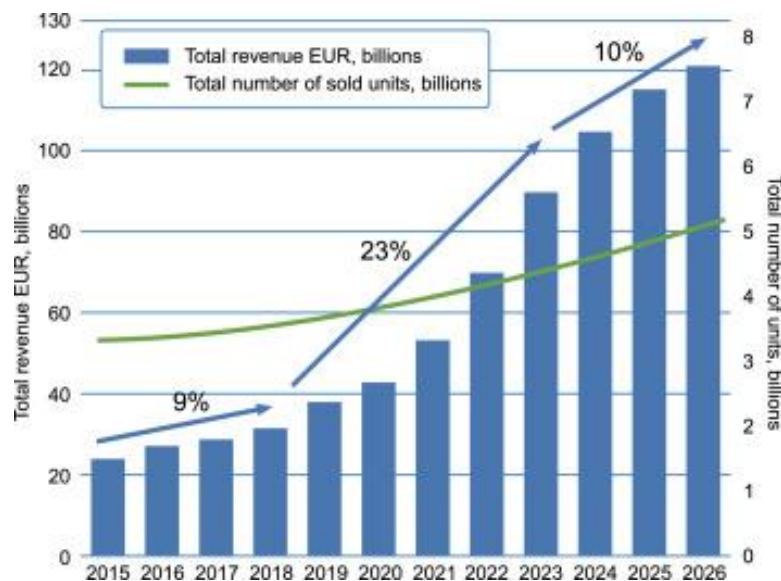


FIGURA 1 Pronóstico de crecimiento del mercado wearable

1.1.1. Características

Las características que definen a un dispositivo para llamarlo *wearable* se determinan en el diseño de la tecnología con la que fueron diseñados, se puede tener conexión con algún otro dispositivo y su funcionamiento permite realizar alguna tarea en concreto.

Uso equitativo

El principio de esta característica es que los dispositivos, compartan un diseño común. El objetivo es brindar la misma utilidad a todos los que usan dichos dispositivos, evitar la división entre usuarios y garantizar la seguridad, la privacidad, el servicio y el diseño para todos los destinatarios, es importante mencionar que estos dispositivos son adecuados para cualquier tipo de persona y se pueden usar sin molestias para cualquier propósito.

Flexibilidad de uso

Las medidas de flexibilidad de estos dispositivos ayudan a adaptarse a la capacidad y precisión del usuario, y permiten refinar y refinar las habilidades de adquisición de datos del usuario, lo que hace que esta función sea más compleja, ya que debe aplicarse al cuerpo. El entorno en el que se encuentra el usuario también se considera importante, ya que las condiciones varían y factores como los niveles de presión e incluso la presión atmosférica son factores importantes.

Uso simple e intuitivo

La facilidad de uso significa que estos dispositivos realmente a pesar de ser portables sean fácil de usar. La complejidad de la tecnología vestible, como debería ser transparente para los usuarios finales. El uso de los dispositivos requiere de la interacción del usuario la cual debe ser mínima para crear una mejor experiencia, en la mayoría se interactúan a través de pantallas pequeñas las cuales facilitan la manera de aprender y usar incluso en un contexto personalizado, lo que significa que la interfaz debe personalizarse por la persona que lo lleva.

Tamaño

El espacio correspondiente que ocupan cada dispositivo debe ser proporcional dependiendo de su propósito, independiente del tamaño del cuerpo del usuario. Las dimensiones de estos

dispositivos suelen estar marcadas con dimensiones únicas, pero debe adaptarse a las personas que lo usarán, esto es una ventaja que ofrecen los dispositivos.

1.1.2. Tipos de wearables

La tecnología vestible ha estado centrada en analizar las actividades físicas de una persona, pero se pueden encontrar varios dispositivos electrónicos vestibles con una gran variedad de funciones. De los diferentes tipos de *wearables* que pueden existir algunos de los más utilizados son:

Wearable Devices para deporte: Son los que permiten medir las variables biológicas, entre las más comunes se encuentran el ritmo cardiaco, calorías consumidas, presión arterial, velocidades de desplazamiento, aceleración, etc. Entre los wearables más utilizados dentro de esta categoría están los smartwatches y los chips instalados en los zapatos de los deportistas.

Wearable Devices para salud: Son los que permiten recopilar información en tiempo real con el objetivo de transmitirla como por ejemplo la glucosa, presión arterial, para que un agente médico pueda monitorear, controlar y diagnosticar alguna eventualidad.

Wearable Devices para Seguridad:

Los dispositivos vestibles también promueven a la seguridad, los cuales permiten registrar las acciones realizadas por los empleados para asegurar que dichas actividades se mantengan con los estándares de seguridad, también existen tecnologías que pueden medir los niveles contaminación, radiación, dióxido de carbono, etc. estos son parámetro crucial Importancia en los sitios de trabajo para evitar los riesgos.

Estilo de vida

Son los que se utilizan en las actividades cotidianas, permitiendo cumplir varias funciones, tal como por ejemplo el reconocimiento de voz. En esta categoría se encuentran los diferentes *wearables* que no tienen un objetivo científico o laboral, sino más bien son sencillamente dispositivos que las personas puedan usar en su día a día como accesorios que les brinden ciertas facilidades en sus actividades diarias.(*The 2017 WT Wearable Technologies*, n.d.)

1.2. FUNDAMENTOS DE MACHINE LEARNING

EL aprendizaje automático *Machine Learning* (ML) es el estudio cuyo objetivo es que las computadoras aprendan a realizar ciertas tareas sin estar programadas para ellos, se le asigna tareas a la computadora y esta las realiza repetidamente es decir adquiere experiencia y así se mide el desempeño en su tarea, cuando se mejora hasta un nivel óptimo se vuelve capaz de generar comportamientos y se puede aplicar de manera confiable en ejemplos nuevos. Crear un programa de aprendizaje automático es diferente del proceso habitual, un programador diseña un algoritmo que toma datos de entrada, aplica reglas y devuelve una salida. El ML se diferencia cambiando modelos y permitiendo que creen sus propias reglas utilizando datos sobre el fenómeno que se está modelando como se observa en la FIGURA 2. El desarrollo de *Machine Learning* se ha motivado por la necesidad de ejecutar tareas repetitivas, riesgosas y que involucren el manejo de una gran cantidad de información las cuales son realizadas deficientemente por los humanos, esto conlleva una mejora en precisión, eficiencia, aprovechamiento de recursos, velocidad de ejecución. Algunas aplicaciones donde se ha aplicado el *Machine Learning* son diagnóstico médico, robótica, reconocimiento facial, reconocimiento de voz, búsqueda de información, control de tránsito vehicular, recomendación de productos, con ello se obtiene una mayor calidad de servicios y productos que consumimos.

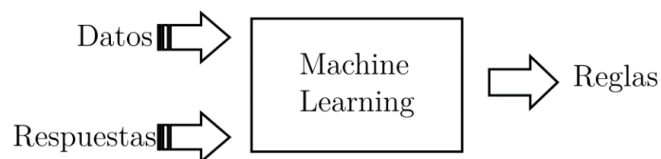


FIGURA 2 Proceso de Machine Learning

1.2.1. Aprendizaje supervisado

El aprendizaje automático se puede clasificar ampliamente en aprendizaje supervisado y no supervisado, el término supervisado significa que la “*máquina*” (el sistema) aprende con la ayuda de algo, por lo general, datos de entrenamiento etiquetados.

Se utiliza para resolver problemas de regresión lineal siguiendo el proceso como lo indica la FIGURA 3 que es una técnica que se genera a partir de números llamados ejemplos de entrenamiento al combinarlos, podemos definir la regresión lineal como una relación entre dos variables donde un aumento en una variable impacta en otra variable para aumentar o disminuir proporcionalmente.(Golden, 2020)

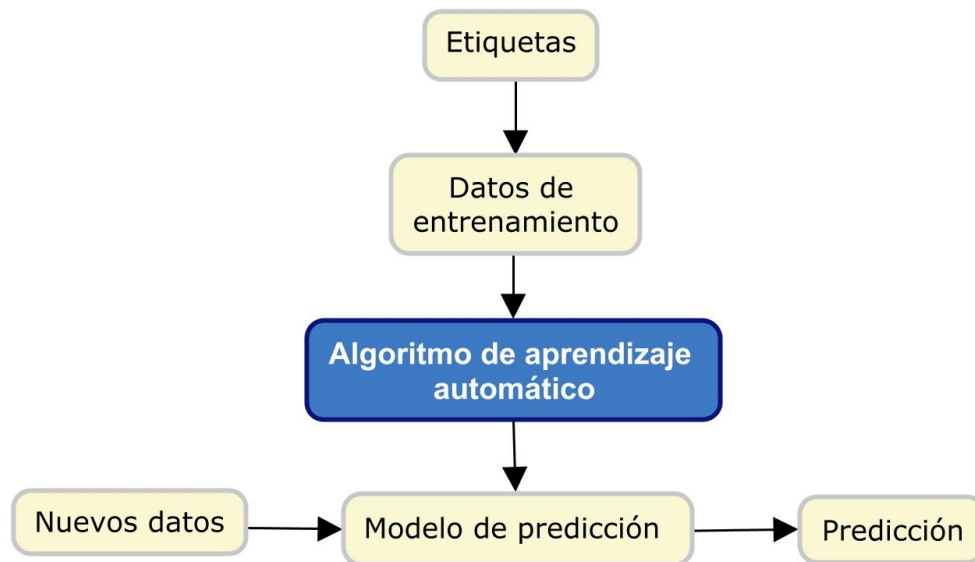


FIGURA 3 Aprendizaje supervisado

1.2.2. Aprendizaje no supervisado

De manera similar, el aprendizaje no supervisado es el proceso de agrupar datos en categorías FIGURA 4. Consiste en analizar información sin etiqueta para encontrar alguna estructura en ella. Hay dos tipos principales de técnicas no supervisadas:

- Enfoque basado en agrupamiento
- Análisis de componentes principales (PCA)

La agrupación en conjunto es un enfoque donde las filas se agrupan, y PCA es un enfoque donde las columnas están agrupadas. Podemos decir que el agrupamiento es útil para asignar un determinado representado en una fila en el conjunto de datos, mientras que

PCA puede ser útil para agrupar columnas, reduciendo la cantidad de variables de los datos. (Golden, 2020)



FIGURA 4 Aprendizaje no supervisado

1.2.3. Aprendizaje reforzado

Este tipo de aprendizaje tiene como objetivo desarrollar un sistema que permite mejorar el rendimiento basado en interacciones con el entorno, incluye una señal de recompensa, a través de esa interacción un agente puede utilizar este tipo de aprendizaje para aprender una serie de acciones que maximicen esta recompensa mediante un enfoque experimental de ensayo error o una planificación de acuerdo a los factores de decisión como se observa en la FIGURA 5. (Python Machine Learning - Vahid Mirjalili, Sebastian Raschka-n.d.)

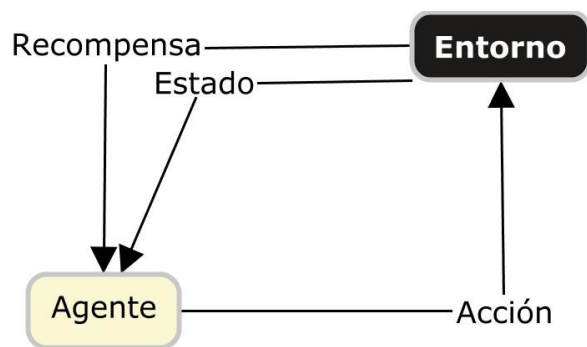


FIGURA 5 Aprendizaje reforzado

1.3. Clasificación de datos

La tarea de clasificar datos es decidir a qué conjunto pertenece un elemento de los datos desconocidos de acuerdo con un grupo de datos. Es por ello que, para facilitar la clasificación, se debe encontrar un modelo que lo represente correctamente considerando los diferentes enfoques para la clasificación de datos. Se debe tomar en cuenta que se tienen que explorar varios algoritmos y analizar el rendimiento obtenido. Los algoritmos que destacan son:

- Logistic Regression.
- K-Neighbors Classifier.
- Decision Tree Classifier.
- Support Vector Machine.
- Linear Regression.

Logistic Regression

La regresión logística ayuda a resolver aquellos problemas, donde hay un número limitado de valores distintos de una variable dependiente. El modelo de regresión logística ECUACIÓN 1, calcula la probabilidad de pertenencia a una clase para una de las dos categorías en el conjunto de datos:

$$p(1|x, \alpha) = \frac{1}{1 + e^{-(\alpha \cdot x)}}$$

ECUACIÓN 1 Logistic Regression

En la FIGURA 6 se observa la regresión logística se aprende de una familia de funciones en un intervalo. Sin embargo, la regresión logística se usa para tareas de clasificación. La clase de hipótesis asociada con la regresión logística es la composición de una función sobre la clase de funciones lineales.

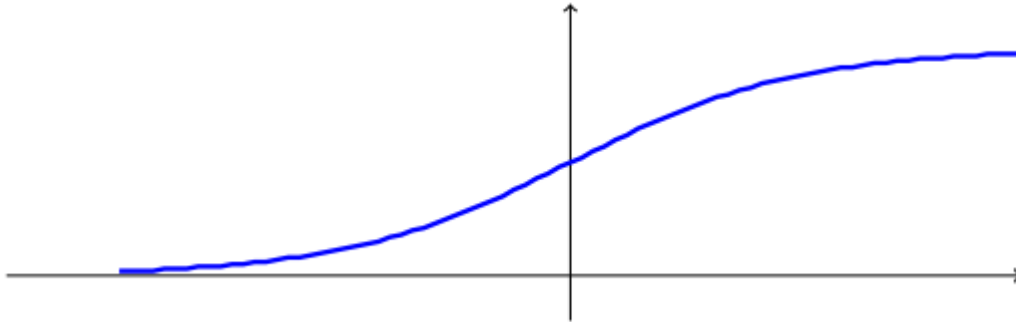


FIGURA 6 Logistic Regression

K-Neighbors Classifier

La clasificación basada en el algoritmo k -vecino usa los datos directamente para la clasificación, sin la construcción previa de un modelo(Dreiseitl & Ohno-Machado, 2002).

La ventaja que tienen los k -vecinos más cercanos sobre otros algoritmos es el hecho de que los vecinos pueden proporcionar una explicación para el resultado de la clasificación más cercano se encuentran entre los más simples de todo el aprendizaje automático

La lógica detrás de tal método se basa en la suposición que las características que se utilizan para describir los puntos de dominio son relevantes para sus etiquetas de manera que los puntos cercanos probablemente tengan la misma etiqueta. Además, en algunas situaciones, incluso cuando el conjunto de entrenamiento es inmenso, encontrar un vecino más cercano se puede hacer extremadamente rápido.(Shalev-Shwartz & Ben-David, 2013)

En la FIGURA 7 se observan los puntos representados límites de una muestra, y la etiqueta predicha de cualquier punto nuevo será la etiqueta del punto de muestra en el centro de la celda a la que pertenece.

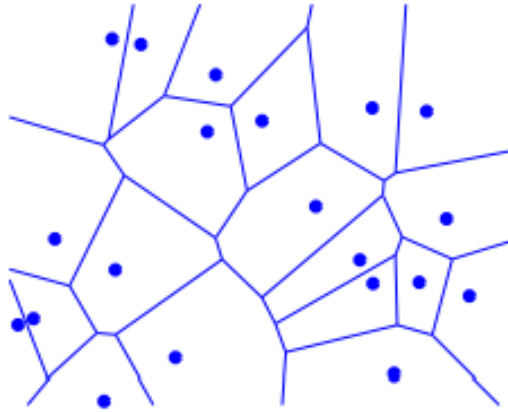


FIGURA 7 Logistic Regression

Decision Tree Classifier

Este algoritmo divide el conjunto de datos varias veces según un criterio que maximiza la separación de datos, lo que da como resultado una estructura de árbol. El criterio más utilizado en este algoritmo es la ganancia de información; lo que esto significa que en cada división se maximiza la disminución de entropía debida a esta división. La estimación de $P(y | x)$ es la proporción de elementos de clase y sobre todos los elementos del nodo hoja que contiene el elemento de datos x . Tiene la característica que se puede expresar fácilmente como reglas por lo que estos modelos se utilizan ampliamente en la salud.(Dreiseitl & Ohno-Machado, 2002). En la FIGURA 8 se observa un árbol de profundidad el camino discontinuo corresponde a la secuencia de predicciones de las hipótesis en las generadas que puede ser 1 o 0.

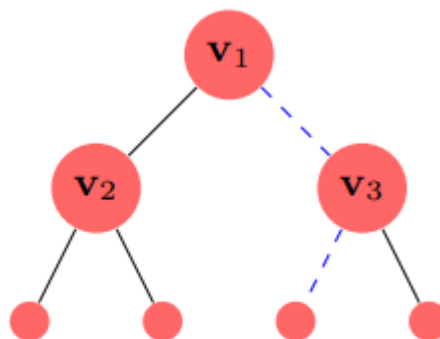


FIGURA 8 Decision Tree Classifier

Support Vector Machine

Este modelo de algoritmo es implementado de la teoría del aprendizaje estadístico, que se encarga del problema de construir estimadores consistentes a partir de datos. Sus algoritmos, construyen límites de separación óptimos entre conjuntos de datos al resolver un problema de optimización cuadrática con restricciones se pueden agregar en el modelo diversos grados de no linealidad y flexibilidad. Debido a que se pueden derivar de ideas estadísticas avanzadas y se pueden calcular los límites del error de generalización, las máquinas de vectores de soporte han recibido un gran interés de investigación considerable en los últimos años.

La desventaja de las máquinas de vectores de soporte es que el resultado de la clasificación es puramente con dos polaridades y no se da ninguna probabilidad de pertenencia a la clase.(Dreiseitl & Ohno-Machado, 2002)

El conjunto de entrenamiento descrito en la FIGURA 9 donde los hiperplanos de color negro discontinuo y verde sólido separan los cuatro ejemplos, se puede decir que es preferible el hiperplano negro sobre el verde. Para determinar esta situación se toma en cuenta el concepto de margen. El margen de un hiperplano con respecto a un conjunto de entrenamiento se define como la distancia mínima entre un punto en el conjunto de entrenamiento y el hiperplano. Si un hiperplano tiene un gran margen, entonces aún separará el conjunto de entrenamiento.

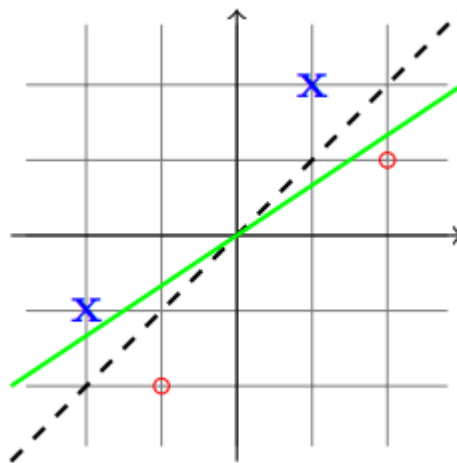


FIGURA 9 Support Vector Machine

Linear Regression

Los modelos lineales son los métodos paramétricos más simples, una regresión lineal es una predicción donde el objetivo es continuo y sus aplicaciones son diversas, por lo que es importante entender cómo un modelo lineal puede ajustarse a los datos, cuáles son sus fortalezas y debilidades, y cuándo es preferible elegir una alternativa, es por ello que siempre merecen la debida atención, porque muchos problemas se pueden resolverse fácilmente con este tipo de modelos (Dreiseitl & Ohno-Machado, 2002). El modelo lineal se basa en la suposición de que es posible aproximar los valores de salida a través de un proceso de regresión basado en la regla de la ECUACIÓN 2.

$$y = \alpha_0 + \sum_{i=1}^m \alpha_i x_i$$

Donde: $A = \{ \alpha_0, \alpha_1, \dots, \alpha_m \}$

ECUACIÓN 2 Linear Regression

En la FIGURA 10 se muestra un ejemplo de un predictor de regresión línea donde $d = 1$, el eje x puede indicar la edad de un bebé, y el eje y su peso, así se puede predecir el peso del bebé en función de su edad y peso al nacer, la clase de hipótesis de predictores de regresión lineal es simplemente el conjunto de sus funciones.

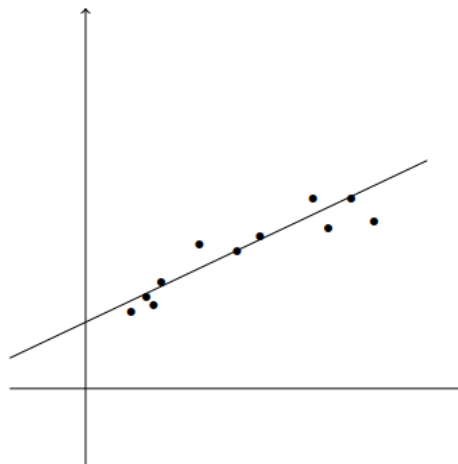


FIGURA 10 Linear Regression

Debido a que existen diferentes algoritmos que realizan las tareas de *Machine Learning*, se elige el algoritmo que mejor se adapte al tipo de acuerdo con los de datos de entrada y de problema que se va a resolver como lo muestra la FIGURA 11.

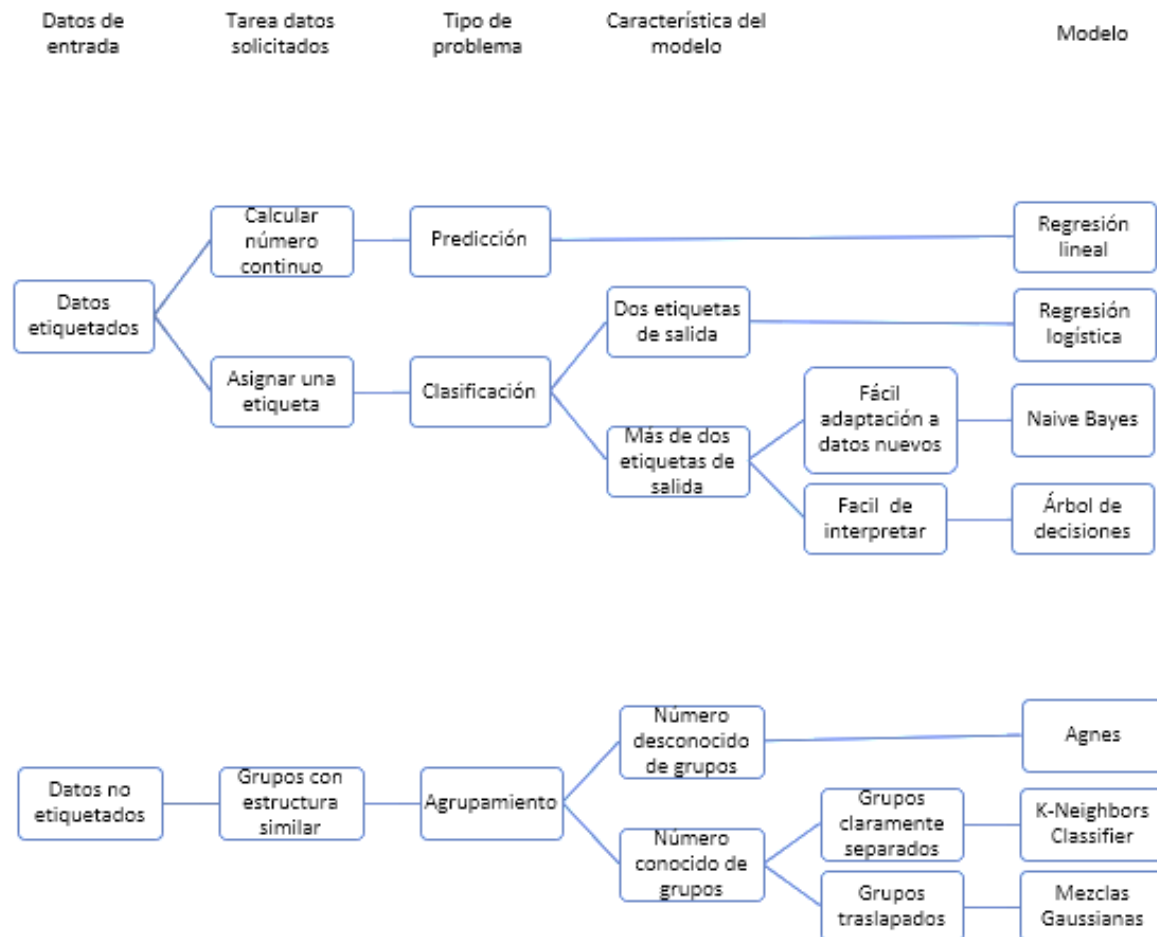


FIGURA 11 Elección del modelo

Microcontroladores y TinyML

Los microcontroladores, tales como las placas Arduino, son sistemas informáticos independientes de bajo costo y cuentan con un solo chip. Por su parte Arduino es una plataforma de código abierto y permite desarrollar aplicaciones de microcontroladores de tal manera que sea accesible para todos. Las razones prácticas por las que es posible implementar ML en microcontroladores, por ejemplo, en la placa Arduino Nano 33 BLE Sense es debido a la función inteligente que actúa rápida y localmente, el costo del hardware es menor, la privacidad de no compartir todos los datos del sensor externamente y la eficiencia del dispositivo

En el lado del aprendizaje automático, existen técnicas que puede usar para ajustar modelos de redes neuronales en dispositivos con memoria restringida como microcontroladores. Uno de los pasos clave es la cuantificación de los pesos de punto flotante a enteros de 8 bits. Esto también tiene el efecto de hacer que la inferencia sea más rápida de calcular y más aplicable a dispositivos de menor frecuencia de reloj.(Situnayake, 2020)

TinyML es un campo emergente y todavía hay trabajo por hacer, pero lo emocionante es que hay un vasto espacio de aplicación inexplorado por ahí. Miles de millones de microcontroladores combinados con todo tipo de sensores en todo tipo de lugares que pueden conducir a algunas aplicaciones TinyML seriamente creativas y valiosas en el futuro.

1.4. Red de sensores corporales

La tecnología de red de sensores corporales, (BSN) juegan un papel cada vez más importante en el área de la salud, el bienestar social y los deportes, además están cambiando la forma en que los humanos usan las computadoras y dispositivos vestibles. La arquitectura de los BSN es de tamaño pequeño y pueden realizar el procesamiento de las señales detectadas por el cuerpo humano para posteriormente comunicarlas al destino requerido donde estos datos pueden usarse para diversos fines. Los BSN están avanzando en la dirección de la inteligencia y la integración de múltiples tecnologías, aunque todavía hay muchos retos por resolver, el futuro de las BSN es bastante prometedor.

1.4.1. BSN

Los BSN se originaron a partir de la red inalámbrica de sensores corporales (WSN), por lo que tienen existen similitudes entre ellos. Sin embargo, las características son correspondientemente diferentes debido a sus diferentes propósitos de aplicación. Los BSN se implementan sobre o alrededor del cuerpo humano, por lo que generan nodos que pueden ser desde pocos hasta algunos de ellos, cada nodo asegura la precisión de los resultados del monitoreo por su la capacidad de adquisición de los datos, debido a las ubicaciones en las que se implementan los nodos, se moverán a medida que se mueve el cuerpo humano y deben tener una alta portabilidad y una compatibilidad.

La arquitectura general de un BSN se observa en la FIGURA 12, los nodos sensores que se colocan en el cuerpo recopilan datos físicos y realizan un procesamiento previo. Los datos son recopilados por un nodo sumidero y luego transmitidos a una estación base para compartirlos, que es la base de muchas aplicaciones, incluidos los sistemas para atención médica, el bienestar social, los pacientes y los servicios de diagnóstico por parte de médicos y expertos. (Lai et al., 2013)

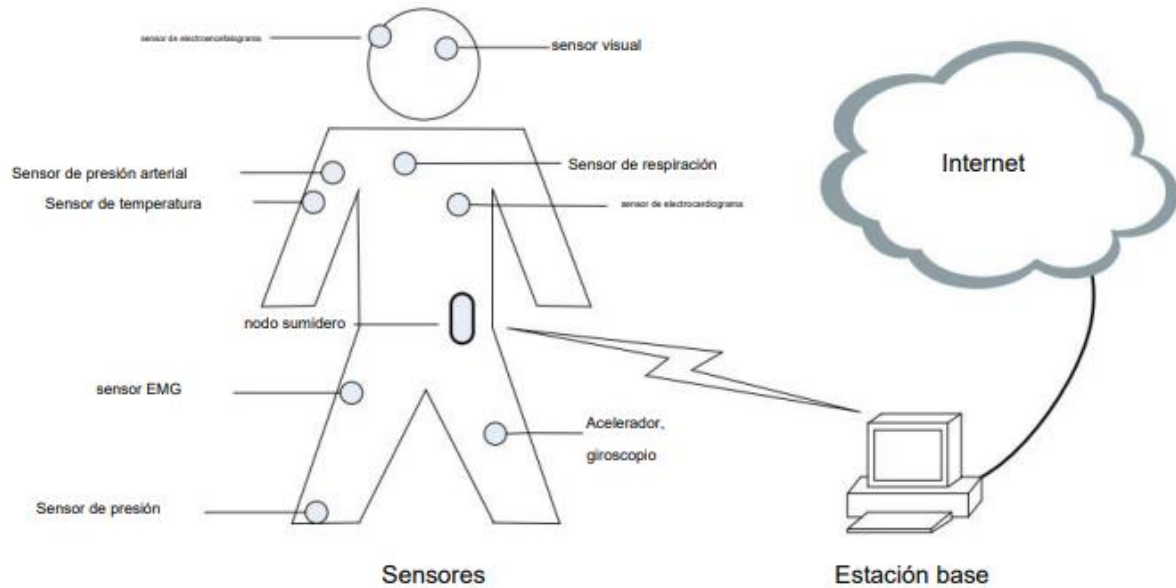


FIGURA 12 Arquitectura de un BSN

Los sensores son aquellos componentes claves para la BSN, se observan los sensores más comunes se observan en Tabla 1 ya que estos permiten conectar el mundo físico con los sistemas electrónicos. Se utilizan principalmente para adquirir información sobre la fisiología y el entorno que los rodea. Los nodos sensores, que tienen un sensor como parte principal, son responsables de procesar la información perspectiva del diseño del sistema, con una explicación detallada del estado de investigación de tres aspectos: sensor, fusión de datos y comunicación de red. Un nodo sensor generalmente se comprende de un módulo sensor, un módulo procesador, un módulo de comunicación y un módulo de fuente de alimentación. El módulo sensor es responsable de adquirir el estado de los datos y convertirlos de señales físicas a eléctricas. En el caso del módulo procesador se encarga de controlar los nodos sensores. Y por último el módulo de comunicación, es responsable de la comunicación entre sensores y computadoras. También se cuenta con el módulo de fuente de alimentación cuya responsabilidad es proporcionar energía para todo el nodo sensor.(Lai et al., 2013)

Tabla 1 Sensores de uso común en BSN (Lai et al., 2013)

Sensores de uso común en BSN			
Sensores	Función	Tipo de señal	Colocación
Acelerómetro	Obtención de aceleración en cada eje espacial del espacio tridimensional.	Continuo	Vestible
Cóclea Artificial	Convertir la señal de voz en pulso eléctrico y enviarla a electrodos implantados en los oídos, generando sensación auditiva al estimular los nervios acústicos.	Continuo	Implantables
Retina artificial	Recibir imágenes capturadas por una cámara externa y convertirlas en señales de pulsos eléctricos, que se utilizan para estimular los nervios ópticos para generar sensaciones visuales.	Continuo	Implantables
Sensor de Presión arterial	Medición de la presión máxima de sistólica y la presión mínima de diastólica	Discreto	Vestible
ECG/EEG/EMG Sensor	Medición de la diferencia de voltaje entre dos electrodos que se colocan en la superficie del cuerpo.	Continuo	Implantables
Giroscopio	Medición de la velocidad angular de un objeto giratorio según el principio de conservación del momento angular.	Continuo	Vestible
Sensor de humedad	Medición de la humedad según los cambios de resistividad y capacitancia causadas por cambios de humedad.	Continuo	Vestible

1.4.2. WBSN

Una red inalámbrica de sensores corporales (WBSN) define a un sistema autónomo que se utiliza para monitorear las actividades de la vida diaria de una persona las WBSN se pueden implementar en un entorno inaccesible, así como también son útiles para detectar enfermedades crónicas como ataques cardíacos, asma, diabetes, etc., pueden funcionar con baterías como el caso del BSN, en el caso de los nodos de sensores en WBSN consumen mucha menos energía.

La arquitectura de la WBSN se observa en la FIGURA 13, los nodos de los sensores se encargan de realizar tres tareas principales: detección de señales, digitalización/codificación/control de señales para comunicaciones que implican acceso múltiple, por último, la transmisión de datos a través de un transceptor de forma inalámbrica. Las señales que se reciben del cuerpo humano no son fuertes y van acompañadas de ruido, para ello se realiza la filtración de la señal para eliminar el ruido. El convertidor analógico a digital convierte las señales analógicas en señales digitales que, después de la digitalización, se almacenan en el microcontrolador.

(Sangwan & Bhattacharya, 2015)

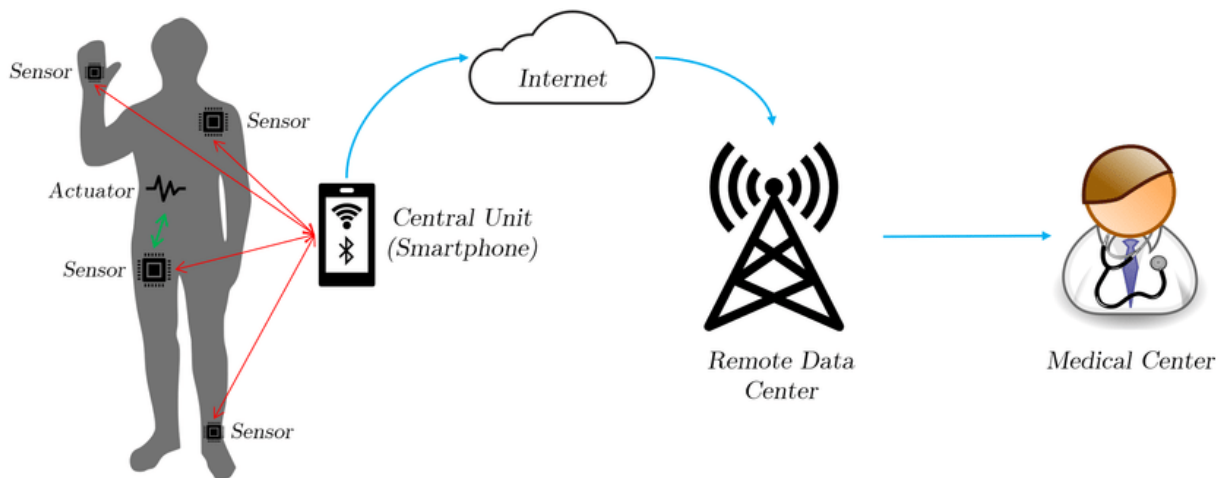


FIGURA 13 Arquitectura de WBSN

1.5. Características estadísticas

Los conjuntos de datos con los que trabajan los dispositivos vestibles pueden contener una gran cantidad de variables o características dentro del contexto del aprendizaje automático es por ello que los datos se deben representar de manera compacta y significativa. Es por ello que se deben extraer características estadísticas en *Machine Learning*, estas características proporcionan información cuantitativa sobre los datos y pueden ser utilizadas como entradas en algoritmos de aprendizaje automático para construir modelos predictivos.

La extracción de características estadísticas puede ayudar a reducir la dimensionalidad del problema al resumir los datos en características más relevantes y significativas. Esto simplifica la tarea de modelo de inteligencia artificial y reduce la complejidad computacional.

Al calcular características estadísticas, se pueden filtrar los valores atípicos o ruidosos en los datos. Esto ayuda a eliminar valores atípicos que pueden afectar negativamente los resultados del modelo y mejorar la calidad de los datos utilizados para el aprendizaje automático.

Otra de las características estadísticas es que permiten resumir la información clave de los datos, como las medidas de tendencia central, la dispersión y la distribución. Estas características proporcionan información cuantitativa que puede ser relevante para el problema de aprendizaje automático en cuestión. Por ejemplo, la media, la desviación estándar o la asimetría pueden ser características útiles en la clasificación o regresión de ciertos tipos de datos.

Al llevar a cabo la extracción de características estadísticas, se busca encontrar propiedades generales que puedan ser útiles en diversos escenarios. Estas funciones se describen a continuación.

La media se calcula sumando todos los valores y dividiendo la suma por el número total de valores. Esta fórmula representa el promedio aritmético de los valores en el conjunto.

La función matemática para calcular la media de un conjunto de valores se define en la ECUACIÓN 3:

$$Mean(x) = \frac{(x_1 + x_2 + x_3 + \dots + x_n)}{n}$$

ECUACIÓN 3 Función Mean

Donde:

$x_1, x_2, x_3, \dots, x_n$: son los valores del conjunto de datos.
n: es el número total de valores en el conjunto.

Harmmean se utiliza para dar más peso a los valores más pequeños del conjunto. Se calcula tomando el recíproco de cada valor, sumándolos y luego tomando el recíproco del resultado.

La función matemática para calcular *Harmmean* de un conjunto de valores se define en la ECUACIÓN 4:

$$Harmmean = \frac{n}{\left(\left(\frac{1}{x_1} \right) + \left(\frac{1}{x_2} \right) + \left(\frac{1}{x_3} \right) + \dots + \left(\frac{1}{x_n} \right) \right)}$$

ECUACIÓN 4 Función Harmmean

Donde:

$x_1, x_2, x_3, \dots, x_n$: son los valores del conjunto.
n: es el número total de valores en el conjunto.

La *Kurtosis* es una medida estadística que describe la forma de la distribución de los datos. Indica el pico de la distribución en comparación con una distribución normal. Una curtosis positiva indica que la distribución tiene colas más pesadas y un pico más pronunciado, mientras que una curtosis negativa indica colas más ligeras y un pico menos pronunciado. La fórmula de la curtosis implica restar la media a cada valor, elevar al cuadrado los resultados, sumarlos, dividir por el número de valores y luego dividir por la cuarta potencia de la desviación estándar.

La función matemática para calcular la *kurtosis* de un conjunto de valores se define en la ECUACIÓN 5:

$$kurtosis(x) = \frac{((x_1 - \mu)^4 + (x_2 - \mu)^4 + (x_3 - \mu)^4 + \dots + (x_n - \mu)^4)}{(n * \sigma^4)}$$

ECUACIÓN 5 Función Kurtosis

Donde:

$x_1, x_2, x_3, \dots, x_n$: son los valores del conjunto.
 μ : es la media (mean) del conjunto de datos.
 σ : es la desviación estándar del conjunto de datos.
 n : es el número total de valores del conjunto de datos.

La fórmula de la *RMS* se utiliza comúnmente para el análisis de las señales y calcular la magnitud efectiva de la señal.

La función matemática para calcular la raíz cuadrática de un conjunto de valores se define en la ECUACIÓN 6 :

$$RMS(x) = \frac{\sqrt{(x_1^2 + x_2^2 + x_3^2 + \dots + x_n^2)}}{n}$$

ECUACIÓN 6 Función RMS

Donde:

$x_1, x_2, x_3, \dots, x_n$: son los valores del conjunto.
 n : es el número total de valores en el conjunto.

La función *Peak2peak* es una medida estadística utilizada para calcular la amplitud o rango total de un conjunto de valores. Esto proporciona una medida de la amplitud total del conjunto, es decir, la diferencia entre el valor máximo y el valor mínimo. La fórmula para calcular el *Peak2peak* se obtiene restando el valor mínimo del conjunto al valor máximo como se muestra en la ECUACIÓN 7.

$$Peak2peak(x) = \max(x) - \min(x)$$

ECUACIÓN 7 Función *Peak2peak*

Donde:

x : es el conjunto de valores.

$\max(x)$: representa el valor máximo de datos.

$\min(x)$: representa el valor mínimo de datos.

La función matemática *Skewness* proporciona información sobre la simetría o asimetría de la distribución de los datos se calcula tomando la media del cubo de las desviaciones de cada valor respecto a la media, y normalizándola por la desviación estándar al cubo. La función matemática para calcular *Skewness* de un conjunto de valores se define en la ECUACIÓN 8.

$$Skewness(x) = \left(\frac{1}{n}\right) * \frac{\sum((xi - \mu)^3)}{\sigma^3}$$

ECUACIÓN 8 Función *Skewness*

Donde:

x : es el conjunto de valores.

xi : son los valores individuales de los datos.

μ : es la media de los datos.

σ : es la desviación estándar de los datos

n : es el número total de valores de los datos.

La función *Trimmean* se utiliza para calcular la media recortada de un conjunto de datos, lo que implica eliminar un cierto porcentaje de los valores más extremos (recortar) y luego calcular la media de los valores restantes. La ecuación matemática para la función *Trimmean* se puede expresar como lo describe la ECUACIÓN 9:

$$\text{Trimmean} = (1 - 2 * k) * \text{mean}(\text{sorted_data}(k + 1 : n - k))$$

ECUACIÓN 9 Función *Trimmean*

Donde:

Sorted_data: es el conjunto de datos ordenados de forma ascendente o descendente.

n: es el número total de elementos en el conjunto de datos.

k: es la cantidad de valores que se recortan en cada extremo (en términos porcentuales).

La varianza y su raíz cuadrada (la desviación estándar), proporcionan una medida más significativa sobre los datos hasta el cual se dispersan las observaciones alrededor de su media. La función matemática para calcular la varianza de un conjunto de valores se define en la ECUACIÓN 10.

$$\sigma^2 = \frac{\sum (X_i - \mu)^2}{N}$$

ECUACIÓN 10 Varianza

Por su parte la Desviación estándar, es la raíz cuadrada de la medida de variación de los valores con respecto a la media. La función matemática para calcular la varianza de un conjunto de valores se define en la ECUACIÓN 11.

$$s = \sqrt{\frac{\sum (X_i - \mu)^2}{N}}$$

ECUACIÓN 11 Desviación Estándar

Donde:

x: es el conjunto de datos.

μ : es la media de los datos.

Σ : es la función de suma.

N: es el número total de elementos en el conjunto de datos.

CAPÍTULO 2. ESTADO DEL ARTE

Como se ha mencionado anteriormente, los dispositivos vestibles y el aprendizaje automático son áreas de gran interés para diferentes sectores los cuales han crecido de manera exponencial en la actualidad. Los dispositivos móviles se han convertido en un vínculo sencillo para el reconocimiento de las actividades humanas, por lo tanto, es común encontrar en cada casa e incluso en cada persona un dispositivo con diferentes sensores como, por ejemplo: temperatura, giroscopio acelerómetro, entre otros los cuales, gracias a los diferentes procesos de adquisición de datos pueden utilizarse con facilidad para capturar los datos que se traducirán en los movimientos corporales. Y todo esto es posible debido a los avances en el desarrollo de los algoritmos de aprendizaje automático.

Los dispositivos vestibles que conocemos y utilizamos actualmente han sido reconocidos por las personas gracias a la introducción en el mercado al marketing de los productos que cada vez toman mayor importancia sin embargo las primeras apariciones de algunos dispositivos están desde el siglo XX de manera general en cuanto a su evolución.

Es así como uno de los primeros dispositivos *wearables* comenzó con la invención de las gafas alrededor del siglo XX por el inglés Roger Bacon (College of Optometrists, n.d. c.1266) Fue así como las gafas de Bacon fueron los primeros wearables diseñados para ser transportados sin problemas y mejorar la visión, convirtiéndose así en las gafas inteligentes.

Otro de los dispositivos vestibles que tuvo una aparición al comienzo del siglo XX fue el primer reloj mecánico de bolsillo, el cual era posible llevar sin ningún problema, elaborado por Peter Henlein. Más tarde, los relojes de bolsillo también se desarrollaron significativamente con la evolución de la miniaturización lo que llevó a la idea de colocar el dispositivo a la muñeca en el siglo XX.

Uno de los momentos más importantes y que impulso a la creación de algunos dispositivos ocurrió durante el período de la Primera y Segunda Guerra Mundial debido a que necesitaban tener comunicación de campo, se crearon los primeros sistemas inalámbricos que llevo al gran avance en la radio portátil.(The National Archives, n.d.)

A mediados del siglo XX un avance significativo en el desarrollo de relojes inteligentes fue la aparición del reloj de calculadora lanzado por Packard(Bensene R, n.d.) el diseño era en miniatura con teclas en la pantalla del reloj. Para finales del siglo la aceleración de los *wearables* se incrementó en la tecnología portátil la cual estuvo a cargo de Nelsonic Space Attacker Watch fue pionero en los juegos portátiles en. El reloj que lanzó estaba equipado con dos botones que permitían jugar juegos de arcade populares en cualquier lugar y en cualquier momento. Después de un año, Nelsonic lanzó Pac-Man portátil y, más tarde, Super Mario Brothers.(Polygon O., p2019)

Para inicios de siglo XXI con la creación de la chamarra hecha de material tecnológico con una red interna diseñada para interconectar dispositivos electrónicos Levi's Industrial Clothing Division (ICD) Jacket lideró el comienzo del siglo, diseñado por Massimo Osti en colaboración con Philips. Este desarrollo se volvió revolucionario para su tiempo e influyó en el desarrollo posterior de marcas como Acronym y Ma. Strum(Kim J.-H., 2007).

La introducción de los primeros auriculares inalámbricos ocurrió en el año 2001 por Plantronics M1000 el cual fue una combinación de un auricular Bluetooth M1000 y un innovador adaptador de teléfono móvil Bluetooth que se conecta directamente a la toma de auriculares, dando a todos los usuarios móviles libertad de auriculares Bluetooth.(Plantronics J.-H., 2010)

Más tarde se lanzó "Nike iPod Sport Kit" un dispositivo que mide y registra la distancia recorrida, el ritmo y otras características de los seres humanos, consiste en un pequeño acelerómetro instalado o ya integrado en los zapatos, que se conecta a un receptor Nike Sportband conectado a los productos Apple.Fitbit fue fundada a principios de por James Park y Eric Friedman, fue el primer rastreador de actividad inalámbrico que podía sincronizar datos con Internet y tener los mismos datos disponibles en un teléfono móvil. (Marshall G., 2016)

La idea de crear un dispositivo en el cual se pudiera mostrar mensajes de teléfonos inteligentes (dispositivos Android y Apple) fue de Eric Migikovsky, de esta manera salió la versión inicial del "smartwatch" reloj que era atractiva por el diseño que era fácil de leer a la luz del día en. El diseño fue muy aceptado por la sociedad del momento. (Marshall G., 2016)

Durante el año 2014 podría describirse como el período donde de los rastreadores de actividad personal se diferenció de otros rangos de acondicionamiento físico al obtener datos como la

frecuencia cardíaca, el consumo de calorías por actividad, varias etapas del sueño y la sudoración, así como la temperatura de la piel con la tecnología Body IQ. El mercado se enfrentó a numerosos proyectos, y el número de personas que los usaban se disparó en solo unos pocos años. (Nguyen M., 2016)

Un invento que surgió en el año 2014 fue la chamarra de energía solar de Tommy Hilfiger para cargar el teléfono. Las baterías solares se cosieron en la chamarra, conectadas a su vez, a la batería, que se encontraba en uno de los bolsillos delanteros. Dos dispositivos podrían estar conectados a la batería, por ejemplo, un teléfono o bien una tableta. Además, los paneles solares podrían desprenderse fácilmente. (Starr M., 2014)

El siguiente avance para los usuarios fue mediante la introducción de Android *Wear* así fue como *Wear OS* fue el primer sistema operativo diseñado específicamente para dispositivos portátiles, particularmente para relojes inteligentes, y fue el comienzo del movimiento de Google hacia la toma de una posición en el mercado de dispositivos portátiles.

Por otro lado, Apple lanzó su primer *wearable*, *Apple Watch* gracias a Steve Jobs quien tuvo que enfrentarse contra el cáncer de páncreas, gracias a esto se dio cuenta que para el personal de enfermería les resultaba difícil monitorear la condición de un paciente ambulatorio y recopilar retrospectivamente la información necesaria sobre el tiempo que pasaba fuera del hospital. Fue entonces cuando Steve Jobs decidió que la atención médica podía mejorarse con la tecnología, y con la ayuda de *Apple* se daba la tarea de resolver el problema de la recopilación de datos. Los rastreadores de actividad existentes no eran adecuados para monitorear las condiciones de los pacientes diariamente. Es por eso que a *Apple* se le ocurrió el *Watch*. Se suponía que la integración con un potente software y una plataforma en la nube facilitaría el trabajo de los médicos. (Bajarin T., 2016)

La clasificación más amplia se basa en el tipo de aplicación de los dispositivos usados y transportados ya que pueden tener una funcionalidad similar, pero factores de forma completamente diferentes, niveles de tecnología, diferentes ubicaciones en el cuerpo, entre otros. Se pueden ver los diferentes tipos en la Tabla 2.

Tabla 2 Clasificación de los tipos de aplicación de los dispositivos wearable.

Tipo	Breve descripción
Funcionalidad de comunicación (C)	Proporciona la posibilidad de no procesar los datos localmente, sino de intercambiarlos con los nodos circundantes y/o la nube remota.
Funcionalidad de control/entrada (CI)	Una amplia área de dispositivos de entrada que van desde botones inteligentes hasta sofisticados dispositivos de reconocimiento de gestos. La tarea principal de este grupo es extender la entrada convencional de interacción hombre-computadora (HCI) centrándose en la usabilidad de los dispositivos manteniendo un factor de forma pequeño como regla.
Educación y deporte profesional (ES)	Tratar de mejorar la educación y la formación mediante el seguimiento de los asistentes.
Tratar de mejorar la educación y la formación mediante el seguimiento de los asistentes.	La mejora de la experiencia de percepción incluye, por ejemplo, sistemas de audio, pantallas de entretenimiento personal, etc.
Heads-up, Información de manos libres (H)	Amplíe las formas convencionales de entrega de datos al usuario utilizando asistentes personales, AR, XR, dispositivos expertos remotos, cámaras portátiles, etc.
Funcionalidad médica/sanitaria (HM)	Separados de los convencionales de detección y monitoreo debido a la necesidad de obtener el estado de dispositivo médico que requiere un esfuerzo significativo en el desarrollo y las pruebas del dispositivo, así como proporcionar un alto nivel de confiabilidad de los datos obtenidos.
Funcionalidad de seguimiento de ubicación (LT)	Requiere tener a bordo algún Sistema Global de Navegación por Satélite (GNSS) o, al menos, una tecnología de comunicación inalámbrica. Por un lado, el concepto aquí corresponde a la conciencia de ubicación desde la perspectiva del nodo y, por otro lado, a la localización remota del dispositivo si es necesario.
Funcionalidad de notificación (N)	Abarca desde una simple notificación de vibración hasta extensiones AR complejas. De manera similar a la funcionalidad de detección, casi cualquier dispositivo personal conectado a la nube directamente o a través de la puerta de enlace puede llevar esta funcionalidad.
Funcionalidad de salida (O)	Varios dispositivos visuales, de audio o habilitados para hápticos para proporcionar al usuario y / o a las personas de alrededor información rápida del ecosistema personal.
Funcionalidad de seguridad y protección (S)	Dispositivos de seguridad personal, asistentes de emergencia, etc.
Funcionalidad de supervisión (M)	Extremadamente sencillo y barato de implementar esta funcionalidad. En general, cualquier dispositivo que tenga un acelerómetro a bordo ya puede proporcionar cierto nivel de detección.
Dispositivos portátiles para mascotas y animales (PF)	Cubre principalmente collares inteligentes, collares de corteza, ropa inteligente, etc.

En áreas como la salud se puede implementar el uso de las BSN, a continuación, se describe el diseño de una red inalámbrica de sensores corporales para la adquisición integrada de señales que se compone de módulos inalámbricos de adquisición de bioseñales, el objetivo del trabajo fue diseñar y desarrollar un sistema inalámbrico *Body Sensor Network* que permite la adquisición sincrónica de la actividad cortical y muscular para la evaluación de las respuestas sensoriales, debido a la arquitectura del sistema y la buena calidad de los resultados obtenidos el dispositivo propuesto representa un avance de la tecnología de punta en cuanto a la adquisición simultánea de señales electroencefalografía (EEG) y superficie de alta densidad EMG (HD-SEMG).

En (McClure et al., 2020) se muestra una investigación donde se desarrolló un sistema de análisis de respiración utilizando datos del acelerómetro y giroscopio implementados en dispositivos vestibles colocados en el pecho y el abdomen para detectar diferentes patrones de respiración para resolver en situaciones médicas de emergencia. Para ello se simuló varios eventos respiratorios y se construyeron conjuntos de datos sintéticos inyectando ejemplos anotados de los diversos patrones en segmentos de respiración normal. Para lograr obtener los resultados se utilizó el modelo de inteligencia artificial red neuronal convolucional para detectar la ubicación de cada evento y clasificarlo en uno de los tipos de eventos anteriores. Se logró una puntuación media del 92 % para respiración normal, 87 % para apnea central del sueño, 72 % para tos, 51 % para apnea obstructiva del sueño, 57 % para suspiros y 63 % para bostezos. Con los resultados de este estudio se demuestra que el uso de los dispositivos vestibles para analizar los datos de movimiento del tórax y el abdomen combinado con los modelos de inteligencia artificial proporciona un medio discreto para monitorear el patrón de respiración y podría tener aplicación en situaciones médicas críticas como la detección de apneas durante el sueño en el hogar y el control de eventos respiratorios en pacientes con ventilación mecánica en la unidad de cuidados intensivos.

En otras áreas como en el deporte donde también se han implementado las BSN es en la investigación realizada por (Sattar et al., 2019) donde se describen los avances que se han realizado con BSN para el análisis del rendimiento en los deportes.

En (Sattar et al., 2019) se describe como los sistemas BSN que utilizan sensores inerciales están sustituyendo a los sistemas basados en vídeo para supervisar el rendimiento en los deportistas, y esto reduce considerablemente el trabajo que presenta la instalación de la cámara de video, así como el procedimiento de calibración. La característica importante del sistema BSN es la manera integrada de crear un dispositivo portátil pequeño y que tenga capacidad de cómputo para procesar los datos utilizando varias técnicas de análisis para monitorear y mejorar el rendimiento en los deportes.

También el trabajo realizado por (Chen, 2021) considera el uso de dispositivos vestibles para evaluar el rendimiento de los atletas durante actividades deportivas. El proceso implica el registro de variables de movimiento a una frecuencia de muestreo lo suficientemente alta a lo largo de una serie de pruebas para monitorear la carga de entrenamiento, capturar e interpretar datos. Para llevar a cabo el análisis propone el uso del algoritmo de aprendizaje supervisado máquinas de vectores de soporte (SVM) para guiar el entrenamiento y monitorear la medición del músculo del atleta y así obtener como resultado las mejoras en las medidas relacionadas con efectos de salud física deficientes. Con esta investigación se pretende promover la investigación colaborativa entre diferentes grupos académicos, clínicas de medicina deportiva y programas de rendimiento de equipos deportivos para ayudar a los jugadores de diferentes campos para utilizar esta tecnología. Además, se ha discutido el uso de un perfil bioquímico complementario y la utilización de dispositivos portátiles para rastrear la inteligencia mental de los atletas. Esto también ha llevado a cabo el desarrollo de nuevos modelos de negocios para proteger a los atletas y ayudarles a medir la protección contra lesiones y problemas de salud y así tener un mejor rendimiento.

CAPÍTULO 3. APLICACIÓN DE LA METODOLOGÍA

3.1. Introducción

En este capítulo se presenta las técnicas investigadas para cumplir con la obtención de datos, la metodología que se utiliza para clasificar las actividades realizadas por los trabajadores se basa en el análisis de las señales de las personas mientras realizan la actividad física de recolección del café, con ayuda de los datos adquiridas por sensores de inercia como el acelerómetro y giroscopio teniendo en cuenta la posición de los puntos de referencia del cuerpo con respecto a los ejes de los sensores. Así una vez obtenidos los datos se debe buscar que tenga la estructura útil para poder utilizarlos, extraer sus características, visualizar las señales y preprocesarlos para poder entrenarlos y desarrollar el modelo de machine a utilizar.

3.2. Análisis de movimientos.

Para realizar un análisis de los movimientos al cortar café es necesario tomar en cuenta varios aspectos, como la postura del cortador, la técnica empleada de corte y la forma de manipular la planta de café.

Para iniciar con la recolección de café se trata de un trabajo físicamente pesado, ya que depende de varios factores especialmente cuando se trata de plantaciones de difícil acceso o a gran altura. Las personas que recogen el café solo llevan un tenate (canasto) que, una vez lleno, se vacía en un costal.

La técnica de corte consiste en hacer un corte limpio y preciso en la rama de café, para evitar dañar la planta y permitir una buena producción en la próxima temporada. Para ello, es importante que el cortador realice el corte correcto de los frutos de café, para ello debe manipular la planta de café con cuidado y delicadeza, evitando movimientos bruscos que puedan dañarla. Es recomendable que el cortador sostenga la rama con una mano mientras realiza el corte con la otra.

La recolección de café es manual y se procede a recolectar desde el tallo del árbol hacia afuera. El movimiento de cortar café es repetitivo. Una vez recolectado el café se debe depositar en el tenate, haciendo cíclico el proceso.

A partir del proceso que siguen las manos durante la recolección, se identifican algunos micro movimientos: Sostener el fruto, Arrancar el fruto con los dedos índice y pulgar y por último soltar el café en el tenate. Los movimientos se describen en la Tabla 3.

Tabla 3 Análisis de movimientos

Movimientos	Descripción
Movimientos en el surco	Realizar el desplazamiento de planta en planta de café. Los recolectores se deben desplazarse en el sentido del surco para cosechar todas las plantas.
Movimientos en la planta	Movimiento en dos etapas: la primera, desde la mitad de la planta hasta la copa de la planta y en la segunda, desde la mitad hacia la parte inferior; finalizar con la recolección del café que se encuentran en el suelo. Se recomienda que el movimiento de los brazos sea en Zigzag.
Micro movimientos de las manos:	El recolector debe buscar, seleccionar, tomar y soltar el café; para evitar el almacenamiento de café en la mano. El recolector debe tomar en cuenta que, al realizar el corte de café, los brazos permanezcan a la altura del corazón.

3.3. Análisis de sensores

Durante los últimos años, con el desarrollo de las BSN en varias áreas de aplicación, los sensores se están convirtiendo en los puntos de investigación más importantes para la obtención de los datos y los algoritmos de procesamiento. Los sensores de inercia (acelerómetros y los giroscopios) se utilizan en las aplicaciones para proporcionar suficiente información y calcular cosas como la dirección, la posición y la velocidad. Para medir estas y otras variables, se combinan dos sensores para crear una unidad de medida inercial (IMU) que proporciona de dos a seis grados de libertad. Las IMU se usan ampliamente en dispositivos que requieren el conocimiento de su ubicación precisa, por ejemplo, robótica, misiles guiados, salud, deportes e instrumentos utilizados en estudios de reconocimiento de actividades humanas.

De acuerdo con los medios para la transmisión de datos, utilizados en la BSN no necesitan consideración de la función de programarse para la obtención de datos, los puede registrar automáticamente configurar, almacenar y así capturar una gran cantidad de datos para procesarlos.

3.3.1. Acelerómetros

La aceleración consiste en saber que tan rápido algo se está acelerando o desacelerando. Y se expresa en unidades de metros por segundo al cuadrado (m/s^2) o fuerza G (g), que es de aproximadamente $9,8 m/s^2$. Es así como los acelerómetros se utilizan para detectar tanto la aceleración estática, es decir la gravedad y la aceleración dinámica cuando existen arranques y/o paradas repentinas. Una de las aplicaciones más utilizadas que se aplican para los acelerómetros es la detección de inclinación. Debido a que se ven afectados por la aceleración de la gravedad, un acelerómetro puede decirte cómo está orientado con respecto a la superficie de la Tierra (Aroganam et al., 2019).. Una de las aplicaciones más comunes se encuentra en los dispositivos móviles, contiene un acelerómetro que le permite saber si se está sosteniendo en modo vertical u horizontal. También se puede usar un acelerómetro para detectar el movimiento. Por ejemplo, se puede usar para detectar golpes de una raqueta de tenis, o balanceos de una bola de boliche. Finalmente, también se puede usar un acelerómetro para detectar si un dispositivo está en estado de caída libre. Esta función se implementa en equipos portátiles que cuentan con discos duros, si se detecta una caída, el disco duro se apaga rápidamente para protegerlo contra la pérdida de datos. Para llevar a cabo estas funciones es importante considerar con qué características cuenta el acelerómetro:

Rango: Los límites superior e inferior de lo que puede medir el acelerómetro también se conocen como su rango. En la mayoría de los casos, un rango de escala completa más pequeño significa una salida más sensible; para que pueda obtener una lectura más precisa de un acelerómetro con un rango bajo de escala completa.

Número de ejes medidos: Se refiere a los tres ejes posibles (x, y y z), Los acelerómetros de tres ejes suelen ser los más comunes para las diferentes aplicaciones.

Uso de energía: Es necesario considerar cuánta energía consumirá el acelerómetro. El consumo de corriente requerido generalmente estará en el rango de $100 \mu A$. Algunos sensores también

cuentan con la función de suspensión para conservar energía cuando no se necesita el acelerómetro.

Funciones adicionales: muchos acelerómetros desarrollados más recientemente pueden tener algunas funciones ingeniosas, más allá de solo producir datos de aceleración. Estos acelerómetros más nuevos pueden incluir características como rangos de medición seleccionables, control del sueño, detección de 0 g.

3.3.2. Giroscopios

Los giroscopios miden la velocidad angular, que es la velocidad a la que algo gira sobre un eje. Si está tratando de rastrear la dirección de un objeto en movimiento, es posible que el acelerómetro no le brinde suficiente información para saber exactamente cómo dirigirlo. A diferencia de los acelerómetros, los giroscopios no se ven afectados por la gravedad, por lo que se complementan a la perfección (Aroganam et al., 2019). Por lo general, verá la velocidad angular expresada en revoluciones por minuto (RPM) o en grados por segundo ($^{\circ}/s$). Los tres ejes de rotación se denominan x, y, z o balanceo, cabeceo y guiñada. El uso de los giroscopios han sido desde tiempos atrás para la navegación espacial, el control de misiles, la guía submarina y la guía de vuelo. En la actualidad ha empezado a utilizarse en conjunto con el acelerómetro para aplicaciones como captura de movimiento de las actividades humanas, aplicaciones en la salud entre otras más.

Rango: Se debe tomar en cuenta la velocidad angular máxima que espera medir que no exceda el rango máximo del giroscopio, para obtener la mejor sensibilidad posible debe tener en cuenta que el rango del giroscopio no sea mayor de lo que espera.

Número de ejes medidos: En comparación con los acelerómetros, los giroscopios están un poco atrasados. Solo han aparecido giroscopios de 3 ejes (x, y, z).

Uso de energía: El consumo de corriente requerido generalmente estará en el rango de 100 μA . Algunos sensores también cuentan con la función de suspensión para conservar energía cuando no se necesita el giroscopio.

Para aplicar la metodología se sigue el diagrama de flujo que se muestra en la FIGURA 14.

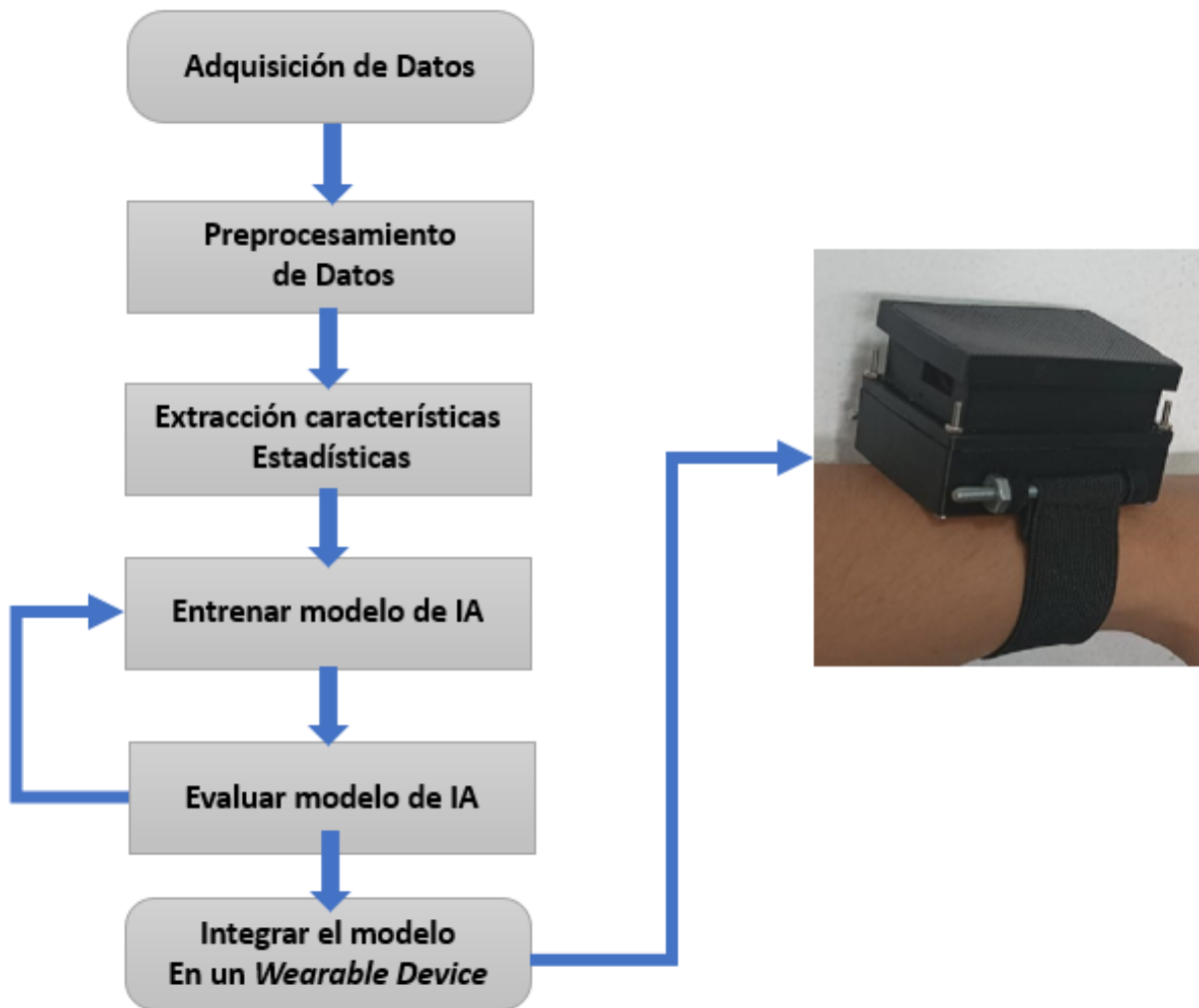


FIGURA 14 Metodología a seguir

3.4. OpenLog Artemis

El OpenLog (OLA), Artemis FIGURA 15 se basa en un sensor registrador de datos de código abierto que esta preprogramado para registrar automáticamente IMU, GPS, datos en serie y varios sensores de presión, humedad y distancia. El OLA, configura y registra automáticamente los datos de los sensores. Se eligió específicamente este sensor debido a que solo se necesitan capturar los datos mediante la unidad de medición inercial (IMU) para el registro integrado de datos con ayuda del acelerómetro, giroscopio y magnetómetro de triple eje.



FIGURA 15 OpenLog Artemis

Dentro del *wearable* device también consta de una batería LiPo a 3.7 volts, FIGURA 16 que tiene como característica principal que es ligeramente delgada y ligera, para proporcionar la energía necesaria del dispositivo vestible.



FIGURA 16 Batería LiPo a 3.7 V

3.5. Diseño

Para la adquisición de datos es necesario tomar en cuenta las actividades que va a realizar la persona en las actividades de recolección de café, es decir cuando la persona se encuentra en de pie o bien cuando la persona inicia algún movimiento para realizar las actividades.

En base al análisis y selección de los dispositivos que reúnen las condiciones necesarias para el sistema se procedió a realizar el diseño del mismo para la adquisición de datos, donde se toma en cuenta de acuerdo a las acciones humanas realizadas en la actividad de recolección del café, para ello se propone el diseño del hardware con la arquitectura del controlador para poder realizar la lectura de los sensores y llevar a cabo su procesamiento de datos siendo la parte más importante del sistema y de esta manera poder clasificar los datos para entregar una respuesta como salida útil para el usuario. Para ello se diseña el modelo en 3D FIGURA 17 del dispositivo vestible que será el encargado de portar la batería para el sensor IMU durante la recolección de datos, se imprime y se ensambla la batería.

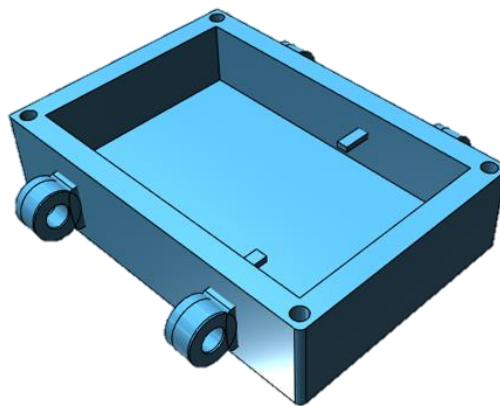


FIGURA 17 Modelo en 3D base para la batería

Para ensamblar el sensor IMU se diseña el modelo de la FIGURA 18, se imprime y se ajusta el sensor sobre el modelo.

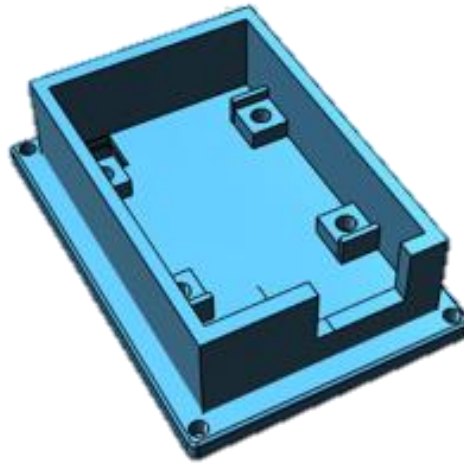


FIGURA 18 Modelo 3D para sensor OLA

Para la parte superior se debe cerrar el dispositivo vestible, para ello se diseña el modelo de la FIGURA 19.

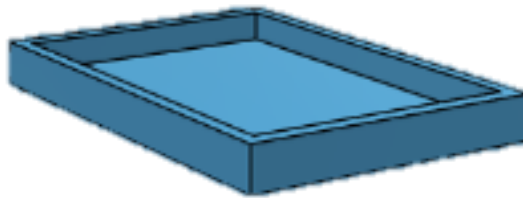


FIGURA 19 Modelo 3D cubierta superior

3.5.1. Posicionamiento de los sensores respecto a la persona

Se define el posicionamiento de los módulos de inercia giroscopio y acelerómetro respecto a la posición de la persona de acuerdo con los ejes mostrados en la FIGURA 20. De esta manera se colocan los módulos sujetos firmemente a las extremidades superior e inferior de la persona, así como al torso del cuerpo. En la FIGURA 21 se observa cuál es la posición relativa de los módulos de inercia respecto al cuerpo de la persona.

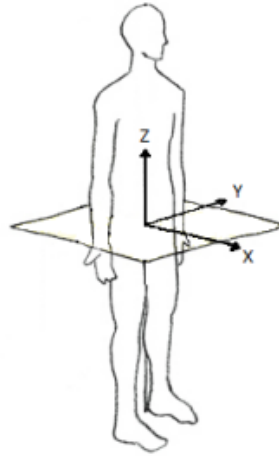


FIGURA 20 Rotación de ejes

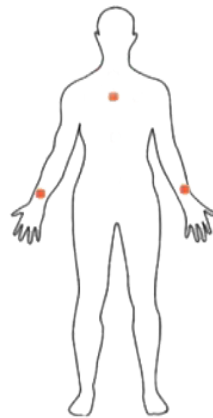


FIGURA 21 Ubicación de Dispositivos Vestibles

3.6. Adquisición de datos

Para entrenar un modelo de clasificación se necesitan datos de entrada para identificar las actividades a clasificar, como primer paso para crear el sistema de clasificación consiste en análisis previo de las actividades humanas que se llevan a cabo durante la recolección de café, por lo que se describen en la FIGURA 22.

En los sistemas de reconocimiento de actividades humanas, para una aplicación específica se crea una base de datos propia que contengan una gran cantidad de datos obtenidos. Para ello, es

importante la elección del tipo de sensor utilizado, así como la ubicación del sensor en el cuerpo, ya que ambos factores de diseño tienen un impacto en los resultados de la clasificación y esto permite generar una base de datos para entrenar dicho sistema.

Actividades a clasificar durante el proceso de recolección de café con el modelo de Machine Learning					
					
<i>STAND UP</i>	<i>WALK</i>	<i>SPU RISE ARM</i>	<i>SPU LOW ARM</i>	<i>LPU RISE ARM</i>	<i>LPU LOW ARM</i>

FIGURA 22 Proceso de recolección de café

Para la adquisición de datos se emplea la BSN sobre los puntos de referencia de los usuarios que son los brazos derechos, izquierdo y sobre el tórax como lo muestra la FIGURA 23.



FIGURA 23 Módulos implementados en el usuario

El objetivo del aprendizaje automático supervisado es construir un modelo que haga predicciones basadas en un conjunto de datos conocidos como entrada y salida para entrenar el modelo y generar predicciones razonables para la respuesta a nuevos datos, por lo tanto, durante las pruebas realizadas se etiquetan las actividades con ayuda de una aplicación desarrollada en Android que permite clasificar cada acción que realice el usuario. Se repiten las actividades hasta que se tengan los datos de cada acción que se va a clasificar. Los datos obtenidos se almacenan en un archivo de texto ya que es fácil trabajar con un formato de archivo plano, como texto o CSV, como lo muestra la FIGURA 24 y facilita la importación de datos para su procesamiento.

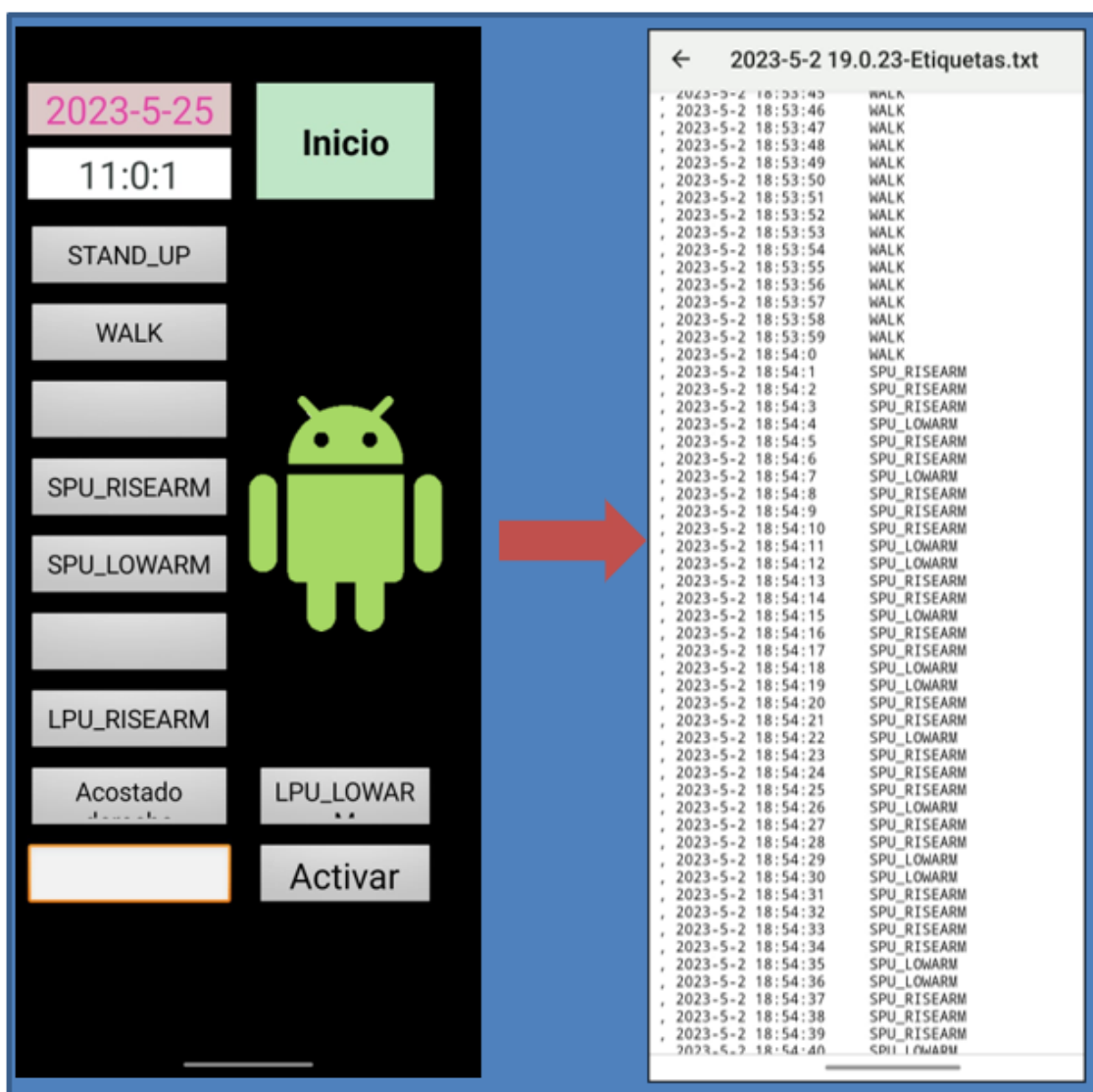


FIGURA 24 Etiquetado de actividades

Para llevar a cabo el reconocimiento de actividades humanas se hace uso de la BSN mediante los dispositivos portátiles, ya que presentan dimensiones y mayor facilidad de colocarlos gracias a las tecnologías actuales, por lo que son ideales para la recopilación de información durante las actividades a realizar y en cualquier entorno. La ubicación en la que se encuentran los dispositivos vestibles sobre el cuerpo durante las actividades a clasificar se observa en la FIGURA 25, de manera que la ubicación del dispositivo en el cuerpo es representativa en las actividades realizadas.



FIGURA 25 Adquisición de datos

Los datos adquiridos son almacenados en un archivo de tipo texto, esto resulta una mayor facilidad para trabajar con un formato CSV como lo muestra la FIGURA 25, y realizar la importación de datos para llevar a cabo el preprocesamiento de las señales obtenidas.

```

Archivo Edición Formato Ver Ayuda
rtcDate,rtcTime,aX,aY,aZ,gX,gY,gZ,mX,mY,mZ,imu_degC,output_Hz,
05/02/2023,18:47:40.37,895.51,84.96,-452.64,-3.41,3.41,-3.35,-81.90,-22.50,-62.25,38.24,250.00,
05/02/2023,18:47:40.42,905.76,146.97,-416.50,-1.22,-7.68,5.06,-82.35,-22.20,-59.70,38.38,35.09,
05/02/2023,18:47:40.43,881.84,104.98,-463.38,-1.83,-7.38,-0.85,-82.35,-23.70,-58.20,38.00,42.86,
05/02/2023,18:47:40.44,791.99,111.33,-482.42,-6.04,-4.33,0.67,-82.50,-23.10,-59.85,38.38,51.28,
05/02/2023,18:47:40.45,804.20,114.26,-475.10,-8.60,1.22,9.51,-81.15,-23.70,-58.95,38.19,58.82,
05/02/2023,18:47:40.46,649.41,208.50,-483.89,-11.16,37.13,13.23,-82.20,-23.85,-58.80,38.24,60.00,
05/02/2023,18:47:40.47,749.02,210.45,-460.94,-5.30,46.16,11.89,-81.60,-23.25,-59.40,38.33,65.42,
05/02/2023,18:47:40.48,846.68,172.85,-386.23,-6.83,55.85,7.93,-81.60,-23.25,-59.40,38.09,69.57,
05/02/2023,18:47:40.49,905.27,138.18,-419.43,-2.44,54.82,6.22,-81.60,-23.25,-59.40,38.33,73.17,
05/02/2023,18:47:40.50,892.09,137.21,-366.70,1.34,49.88,5.79,-82.20,-21.60,-59.10,38.38,76.34,
05/02/2023,18:47:40.51,869.63,162.11,-362.30,0.06,34.70,1.59,-82.20,-24.30,-58.50,38.05,75.86,
05/02/2023,18:47:40.52,870.12,166.02,-358.89,3.48,30.98,6.71,-82.20,-24.30,-58.50,38.38,78.43,
05/02/2023,18:47:40.53,852.05,168.95,-367.68,4.88,26.59,4.76,-83.40,-22.50,-58.20,38.33,81.25,
05/02/2023,18:47:40.53,836.43,166.99,-375.98,6.65,26.40,1.04,-82.65,-24.45,-59.40,38.19,83.33,
05/02/2023,18:47:40.54,826.17,166.50,-389.65,7.50,28.17,-0.73,-81.00,-22.35,-58.80,38.19,85.23,
05/02/2023,18:47:40.55,818.36,160.16,-388.18,11.10,30.55,-1.40,-81.00,-22.35,-58.80,38.24,87.43,
05/02/2023,18:47:40.56,820.31,152.83,-403.32,14.39,37.26,-5.12,-82.50,-23.85,-58.50,38.00,85.86,
05/02/2023,18:47:40.57,815.43,143.55,-382.32,10.24,37.80,-2.99,-82.50,-23.85,-58.50,38.09,87.38,
05/02/2023,18:47:40.58,833.98,162.11,-378.91,11.95,42.44,-4.15,-82.50,-23.85,-58.50,38.19,89.20,

```

FIGURA 26 Archivo CSV

3.7. Preprocesamiento

La mayoría de los conjuntos de datos necesitan algún procesamiento previo antes de la extracción de características; las tareas incluyen la eliminación de valores atípicos y tendencias ya que los algoritmos de aprendizaje automático no son lo suficientemente inteligentes como para diferenciar entre el ruido y la información valiosa, es por eso que antes de usar los datos para el entrenamiento se debe asegurar que los datos estén limpios y completos.

El objetivo principal de esta etapa es preparar y limpiar los datos de manera adecuada para garantizar que sean aptos y representativos para el proceso de entrenamiento del modelo.

Para llevar a cabo el preprocesamiento de datos se utiliza el software MATLAB®, el cual implica una serie de pasos para preparar y limpiar los datos antes de aplicar algoritmos de aprendizaje automático, la FIGURA 27 proporciona una descripción general de los pasos comunes para el preprocesamiento de datos en MATLAB®.

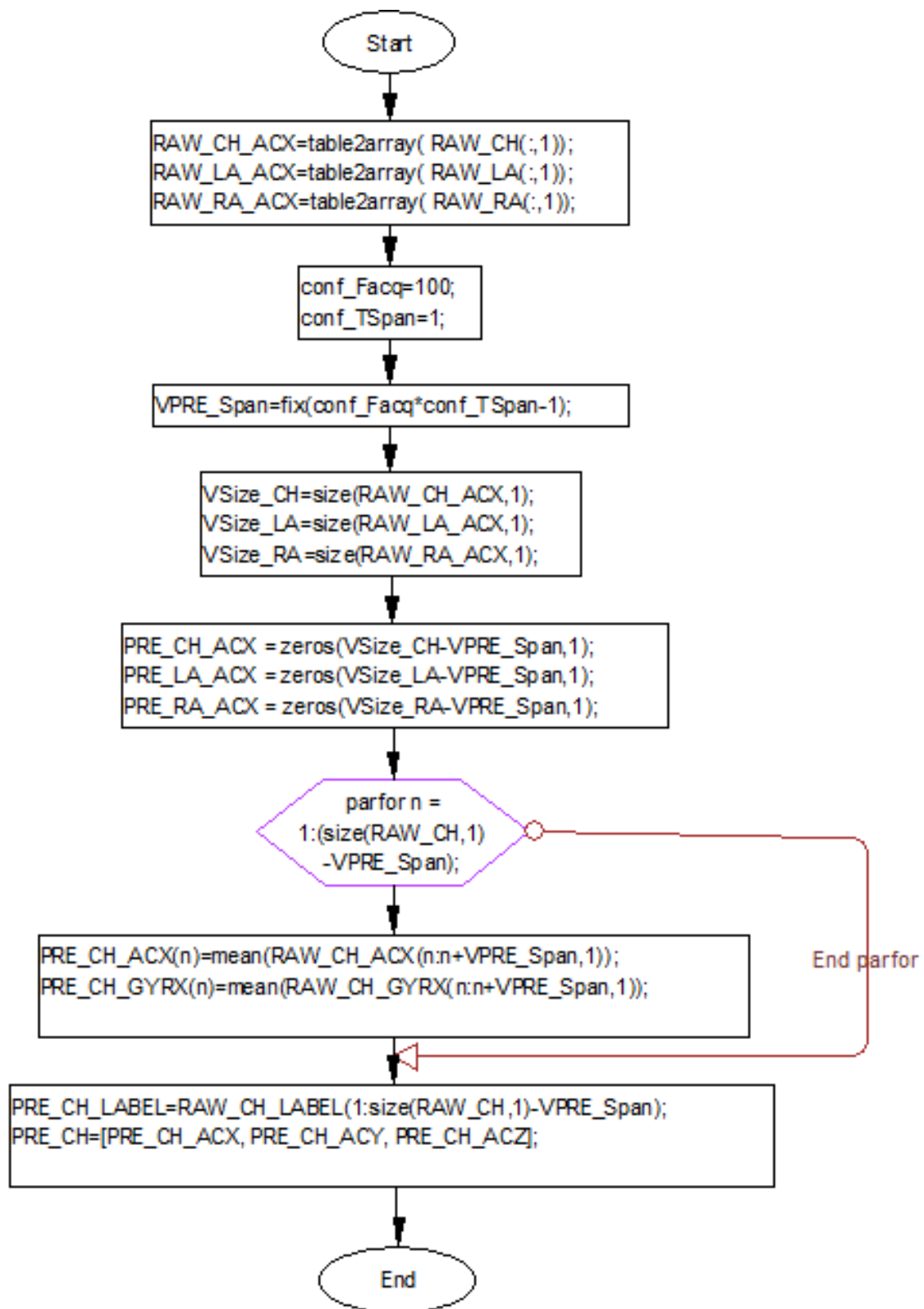


FIGURA 27 Preprocesamiento de datos

Para llevar a cabo esta etapa se debe disponer de la base de datos para poder entrenar un modelo clasificador de datos, pero para ello los datos obtenidos por los dispositivos vestibles deben ser filtrados con el objetivo de reducir el ruido que se genera en la medición del sensor. El filtrado se lleva a cabo mediante una segmentación de la señal para permitir obtener pequeños periodos de señal donde se puede detectar una cierta actividad en un periodo determinado. De este modo, se consigue extraer ciertas características más discriminativas de cada actividad para poder llevar a cabo su clasificación, por lo que se lleva a cabo mediante el método de ventaneo temporal como lo indica FIGURA 28 donde únicamente se analizan pequeños partes de la señal mediante una ventana temporal donde la anchura está definida dependiendo la frecuencia con la que se obtuvieron los datos y así se va deslizando lo largo de toda la señal. En la FIGURA 29 se observa se logra observa la señal procesada eliminando los valores atípicos.

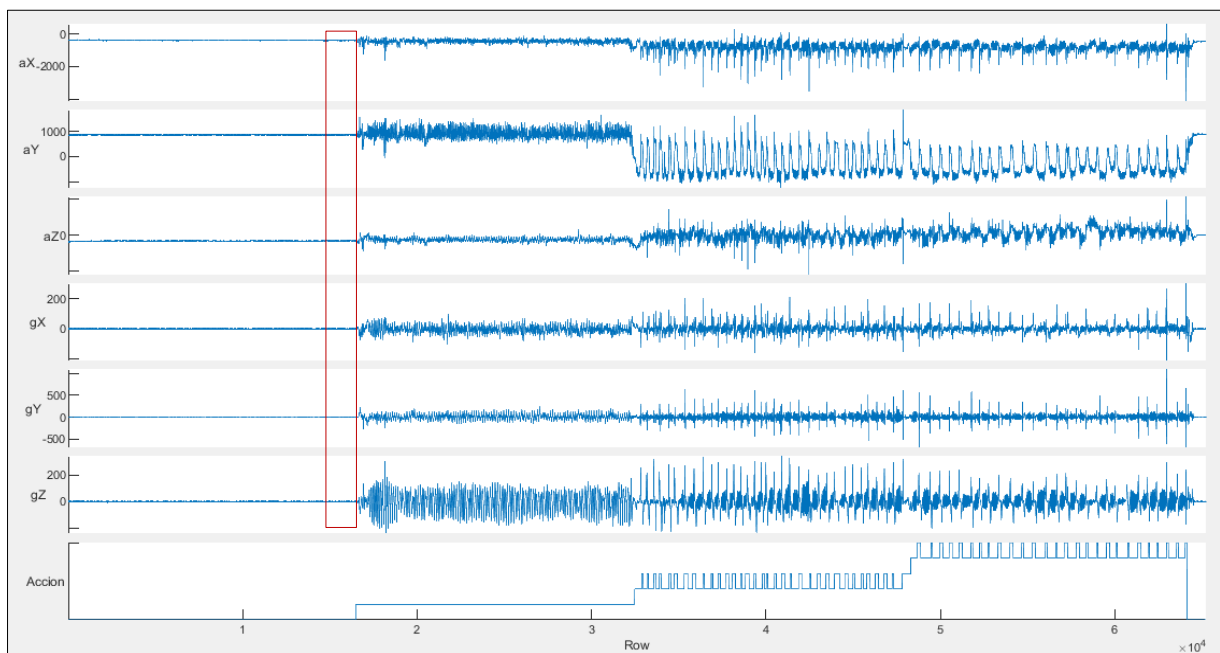


FIGURA 28 Ventana temporal deslizante

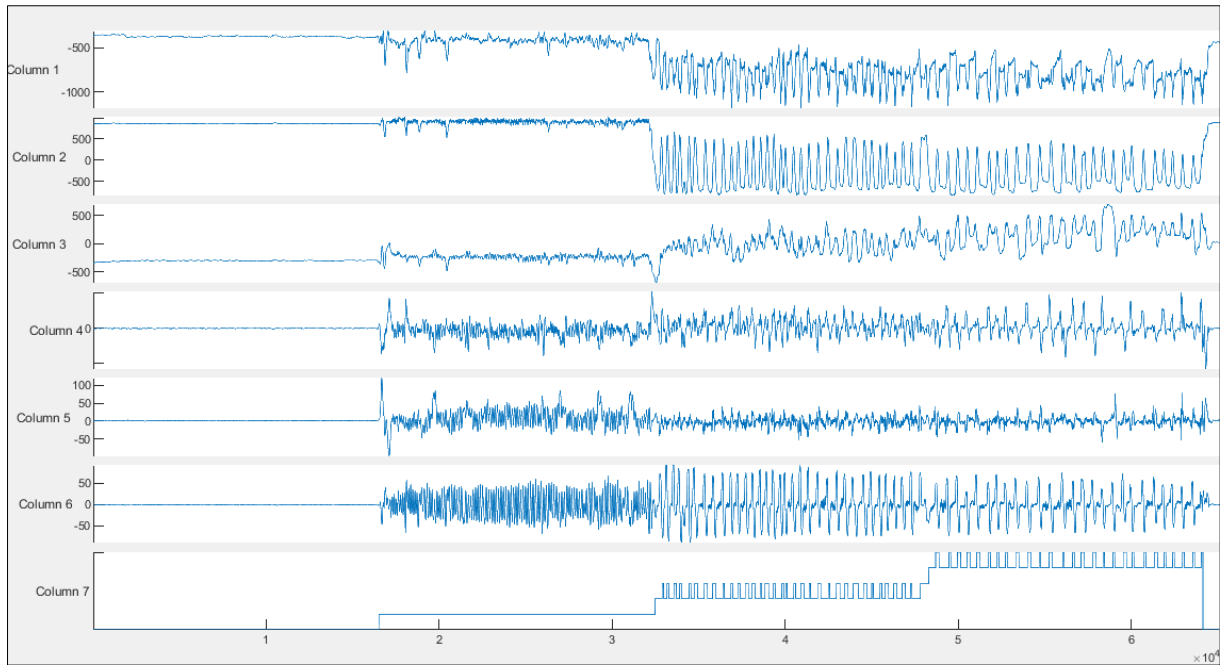


FIGURA 29 Señal preprocesada

3.8. Extracción de características

Para la fase de extracción de características de la base de conocimiento consiste en hacer uso de estas características como datos de entrada para un clasificador que permite mejorar los resultados obtenidos en la clasificación, en comparación si estos se utilizan directamente de los datos obtenidos por el dispositivo vestible.

Las características obtenidas son representativas de acuerdo con el dominio temporal y frecuencial que se toma en cuenta y permiten crear diferentes grupos para comprobar con cuál de ellos se logra obtener un mejor resultado cuando se clasifican. Las características comúnmente más utilizadas se muestran en la Tabla 4.

Tabla 4 Características seleccionadas

Características seleccionadas	
Mean	Devuelve la media de los elementos de la primera dimensión de matriz
Harmmean	Calcula la media armónica de una muestra.
Kurtosis	Devuelve la muestra de curtosis
Median	Devuelve el valor medio
Mode	Devuelve el valor que aparece con más frecuencia
Min	Devuelve los elementos mínimos de una matriz.
Max	Devuelve los elementos máximos de una matriz
“Peak2Peak”	Devuelve la diferencia entre los valores máximo y mínimo
Rms	Devuelve el valor de la raíz cuadrada media (RMS) de los datos de entrada
Skewness	Devuelve la asimetría de la muestra
Std	Devuelve la desviación estándar de los elementos.
Sum	
Trimmean	Devuelve la media de los valores
Var	Devuelve la varianza de los elementos

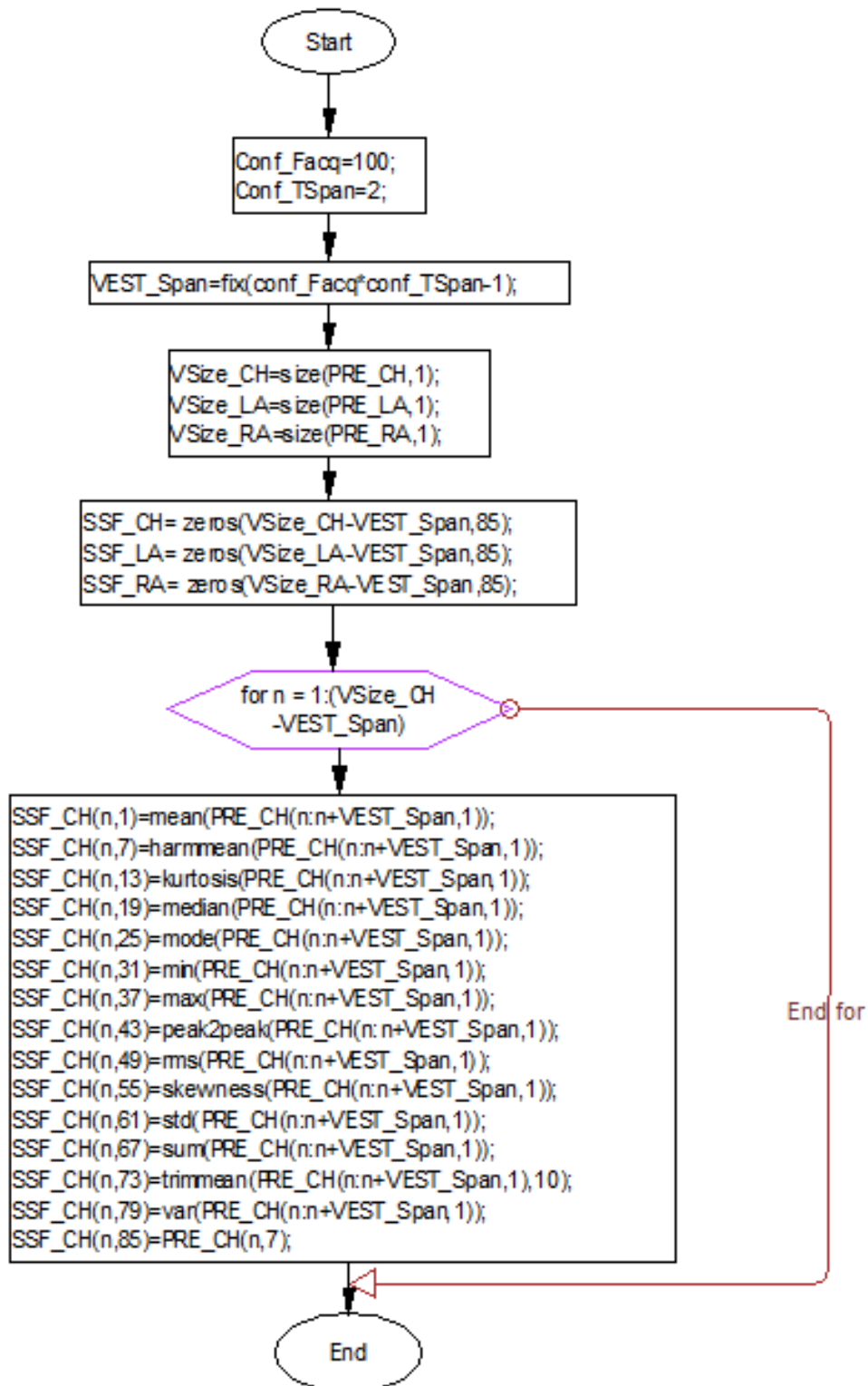


FIGURA 30 Características Estadísticas

En la FIGURA 30 se observa la extracción de características mediante el software de aplicación MATLAB®. La extracción de características produce mejores resultados en la señal obtenida se muestra en la FIGURA 31 para aplicar al aprendizaje automático directamente a los datos.

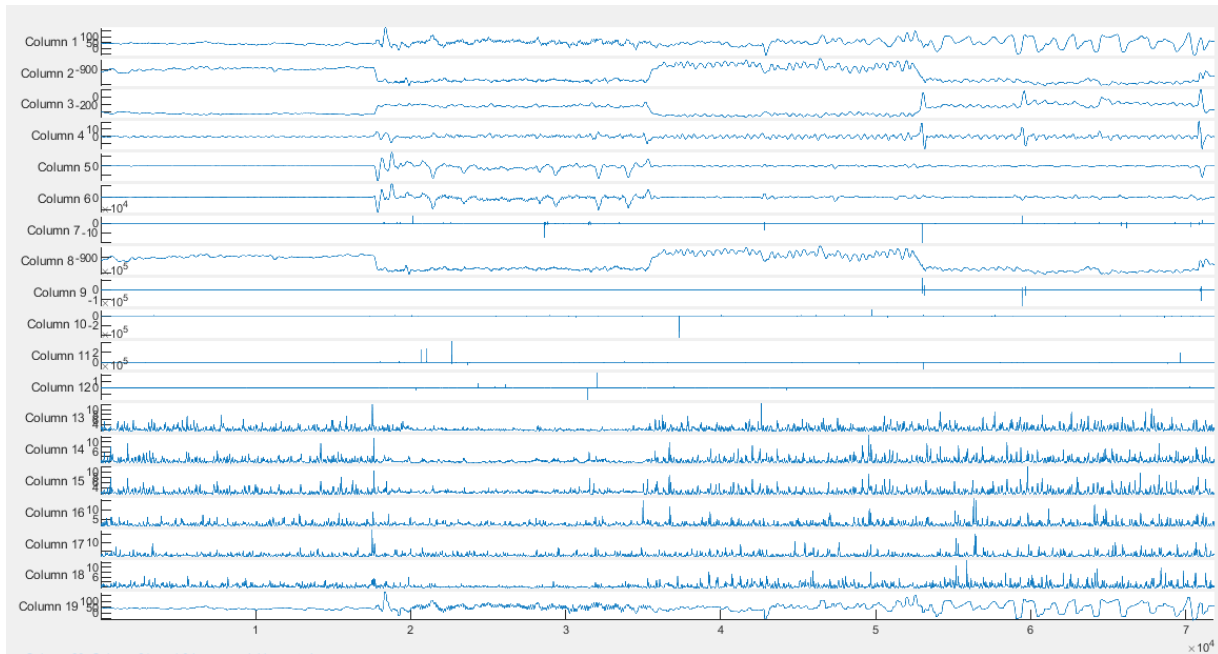


FIGURA 31 Señal con características estadísticas

3.9. Creación del modelo.

Para llevar a cabo el modelo de *Machine Learner*, se realiza con la aplicación *Classification Learner* del software de MATLAB®, en la FIGURA 32 el cual se encarga de entrenar modelos para clasificar datos. Con esta aplicación, se puede explorar el aprendizaje automático supervisado utilizando varios clasificadores, explorar los datos, seleccionar sus características, especificar esquemas de validación, entrenar modelos y evaluar los resultados. Para realizar el entrenamiento automatizado se lleva a cabo con el modelo de clasificación árbol de decisión en la FIGURA 33, debido a que el aprendizaje automático es supervisado ya que proporciona un conjunto de datos etiquetados de las actividades que se van a clasificar.



FIGURA 32 Classification Learner

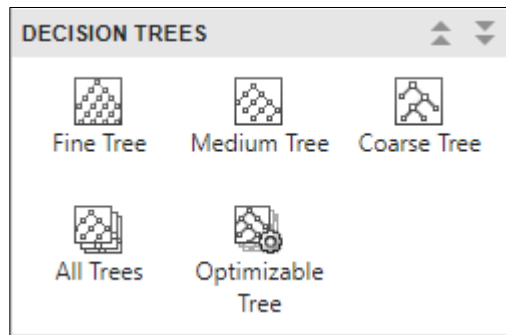


FIGURA 33 Decision Trees

New Session from Workspace

Data set

Data Set Variable

SSF_CH 71860x85 double

☒ Use columns as variables
☐ Use rows as variables

Response

☒ From data set variable
☐ From workspace

Predictors

	Name	Type	Range
☒	column_1	double	0 .. 0
☒	column_2	double	0 .. 0
☒	column_3	double	0 .. 0
☒	column_4	double	0 .. 0

Add All Remove All

[How to prepare data](#) Refresh

Validation

Validation Scheme

Cross-Validation

Protects against overfitting. For data not set aside for testing, the app partitions the data into folds and estimates the accuracy on each fold.

Cross-validation folds 0

[Read about validation](#)

Test

☐ Set aside a test data set

Percent set aside 10

Use a test set to evaluate model performance after tuning and training models. To import a separate test set instead of partitioning the current data set, use the Test Data button after starting an app session.

[Read about test data](#)

Unable to find suitable response variable in the data set. Select another data set or choose a response from the workspace.

Start Session Cancel

FIGURA 34 Workspace

Para configurar el espacio de trabajo se utiliza la FIGURA 34, se debe seleccionar la tabla que contiene las variables (Data set Variables) con el conjunto de datos preprocesados, considerando un Para iniciar con la validación predeterminada se presiona la opción Start Session.

La aplicación muestra un resumen en la FIGURA 35, con resultados del modelo, la precisión del modelo y los parámetros obtenidos, posteriormente se cargará el diagrama y el porcentaje de efectividad, este tipo de algoritmo potencian los modelos con una alta precisión, estabilidad y facilidad de interpretación, son adaptables a cualquier tipo de problema ya que tienen la estructura de un diagrama de flujo, utilizan nodos y divisiones en su desarrollo.

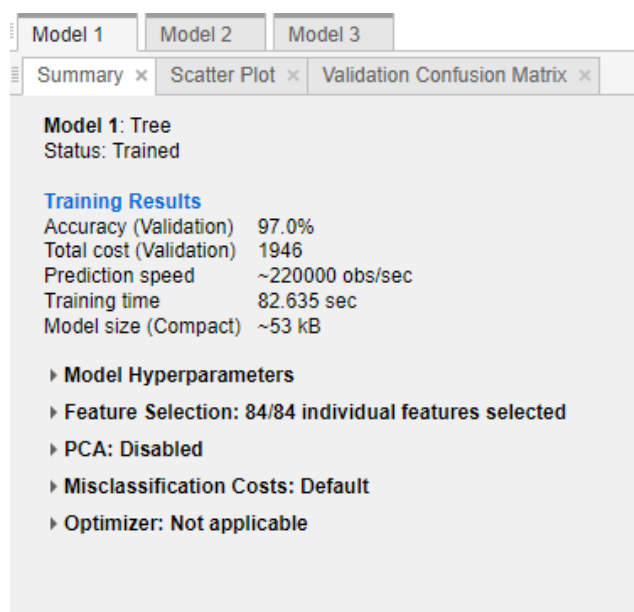


FIGURA 35 Summary

En la FIGURA 36 se muestra la gráfica de dispersión la cual permite identificar los predictores que separan las clases trazando diferentes pares de predictores en la gráfica de dispersión. Con esta grafica se puede visualizar los datos de entrenamiento y los puntos mal clasificados en el en el modelo.

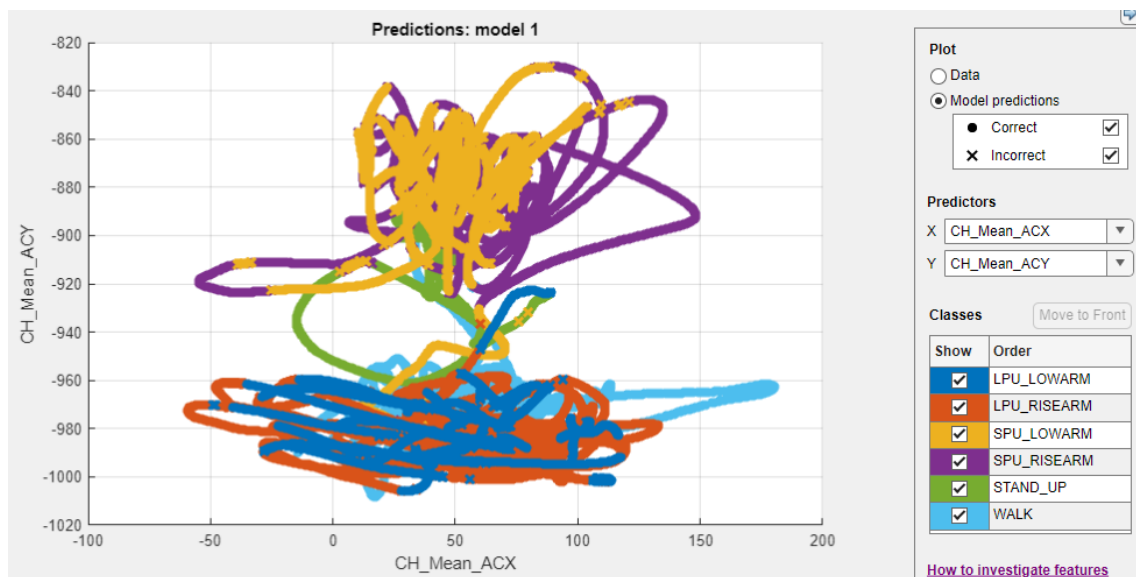


FIGURA 36 Prediction Model

En la FIGURA 37 se observa la matriz de confusión la cual tiene como función comprender cómo se desempeñó el modelo cada una de las actividades a clasificar. Las celdas diagonales muestran dónde coinciden la clase verdadera y la clase predicha de cada actividad

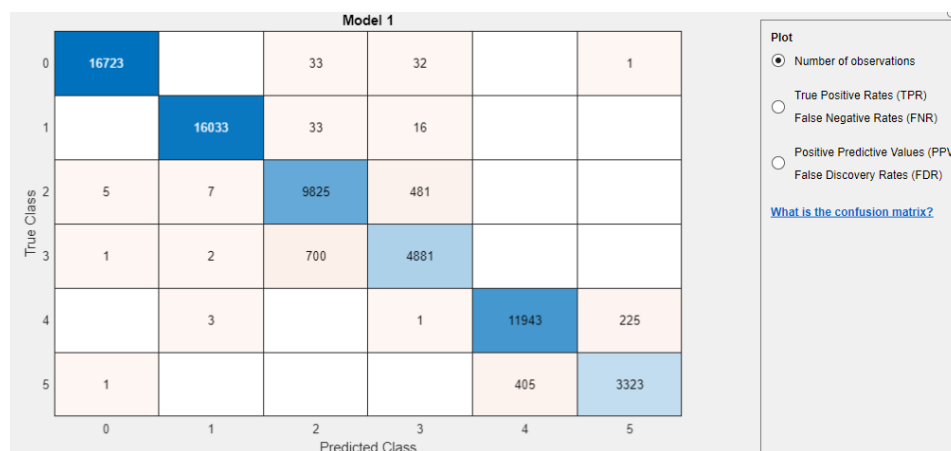


FIGURA 37 Confusion matrix

La FIGURA 38 muestra curva ROC la cual tiene como función mostrar el valor de las tasas positivas verdaderas obtenidas indicando que el modelo clasificador asigna correctamente las clasificaciones de las actividades.

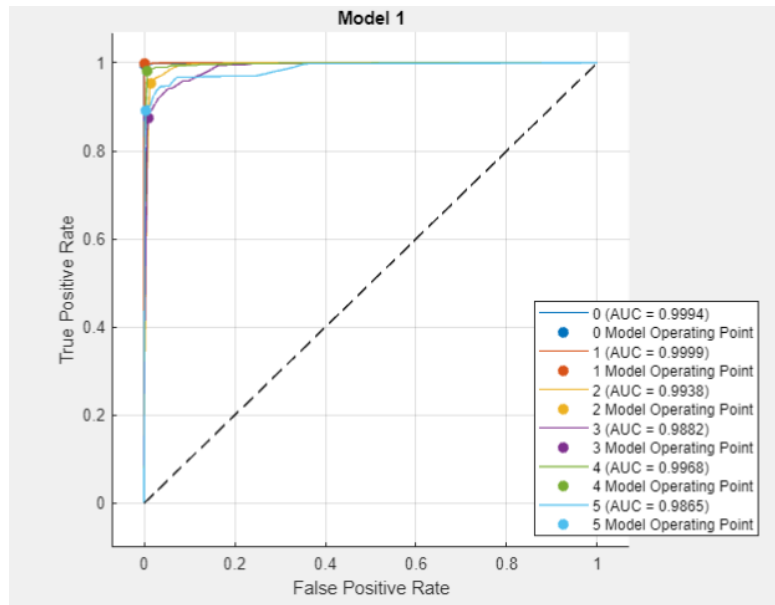


FIGURA 38 Validation (ROC) Curve

Los árboles de decisión tienen la función de predecir las respuestas a los datos entrenados. Para predecir una respuesta, se debe seguir la decisión que evalúa el árbol desde el nodo raíz (comienzo) hasta un nodo hoja. El último punto contiene la respuesta. Para ellos se debe entrenar el modelo utilizando *Tree*, *Coarse* como lo muestra la FIGURA 39, que es uno de los modelos de clasificación más simples de interpretar, pero con un bajo nivel de clasificación.

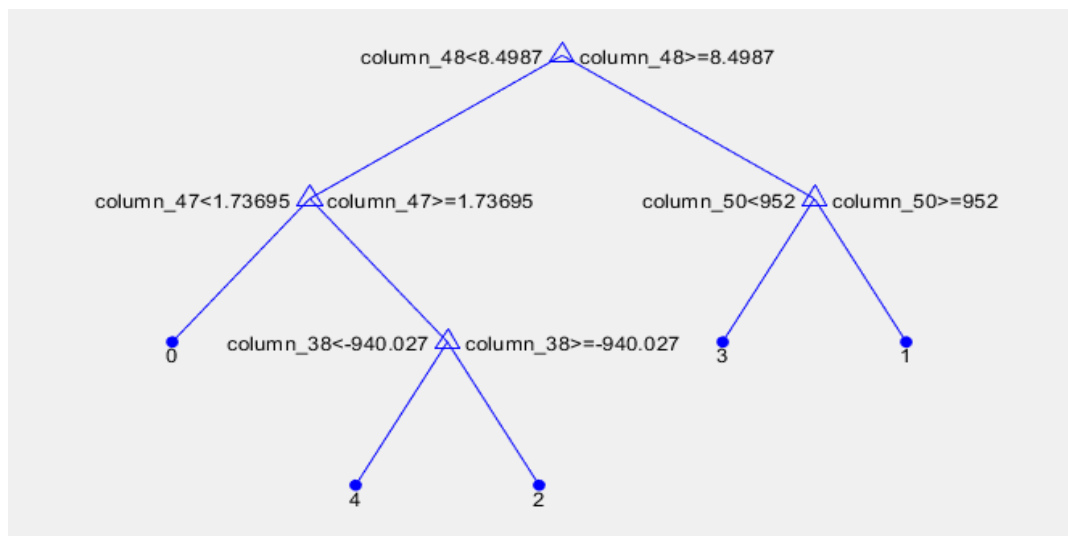


FIGURA 39 Coarse Tree

En la FIGURA 40 se muestra el árbol de decisión *Tree Medium*, el cual muestra una clasificación de actividades con un mayor índice de predicciones en el modelo.

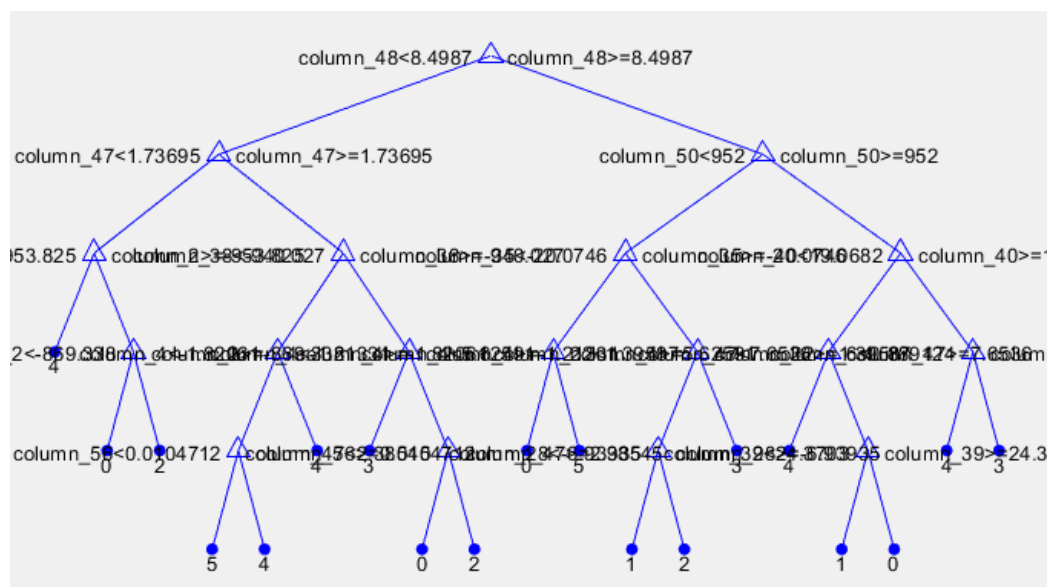


FIGURA 40 Medium Tree

En la FIGURA 41 se genera el *Fine Tree* con una validación del modelo de inteligencia artificial confiable ya que el porcentaje de predicción se logra observar en la cantidad de nodos que contiene las respuestas de las actividades clasificadas.

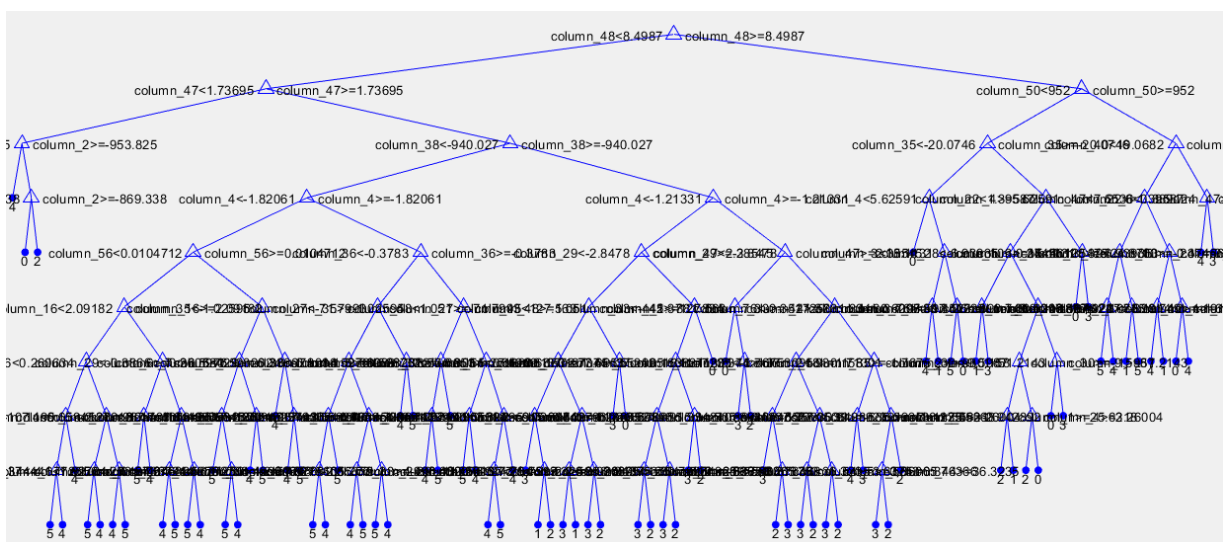


FIGURA 41 Fine Tree

CAPÍTULO 4. RESULTADOS Y DISCUSIÓN

Una vez ejecutados los algoritmos de *Machine Learning* en el software de MATLAB®, estos crearán los diferentes grupos de modelos obtenidos y comenzará la evaluación de los distintos modelos de clasificación para la posterior aplicación de los algoritmos que ofrece las mejores predicciones. Para ellos se analizan los modelos de inteligencia artificial obtenidos de los módulos *Chest*, *Left Arm* y *Right Arm*.

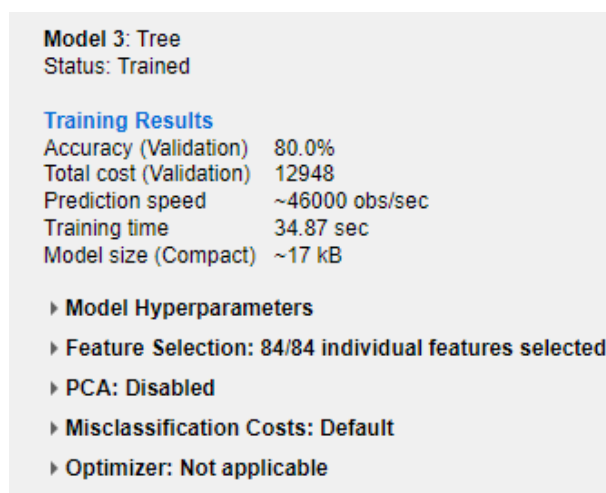


FIGURA 42 Summary Coarse Tree CH

4.1. Modulo Chest

Los árboles de decisión, o árboles de clasificación se caracterizan por predecir respuestas a los datos entrenados, para iniciar con la obtención de los modelos se entrenó el algoritmo de inteligencia artificial *Coarse Tree* y en los resultados se logra observar que no cuenta con una alta precisión de entrenamiento. El sentido de que su precisión de entrenamiento es baja a la de un conjunto de prueba representativo, no logra clasificar todas las actividades de la persona cuando se encuentra realizando la recolección de café. En el resumen obtenido se puede ver las métricas del modelo para evaluar y comparar los modelos generados. Las métricas de resultados de entrenamiento se observan en FIGURA 42, calculan el conjunto de validación y muestra el porcentaje obtenido con un 80% de efectividad, la velocidad de predicción, el tiempo empleado

para el entrenamiento y el tamaño del modelo para su exportación, por lo que se puede observar que sus características son rápidas pero su nivel de entrenamiento bajo.

El gráfico de dispersión se utiliza para examinar los resultados del clasificador, su función es mostrar las predicciones del modelo. En otras palabras, el software obtiene cada predicción utilizando el modelo que fue entrenado sin la observación correspondiente. Para observar los resultados se pueden realizar seleccionando las características a mostrar en los ejes X e Y de los predictores como se muestra en la FIGURA 43, visualizando los resultados por clase mostrando u ocultando clases específicas usando las casillas de verificación de datos correctos e incorrectos, por lo que en la se figura se logra observar que indica un porcentaje bajo de efectividad.

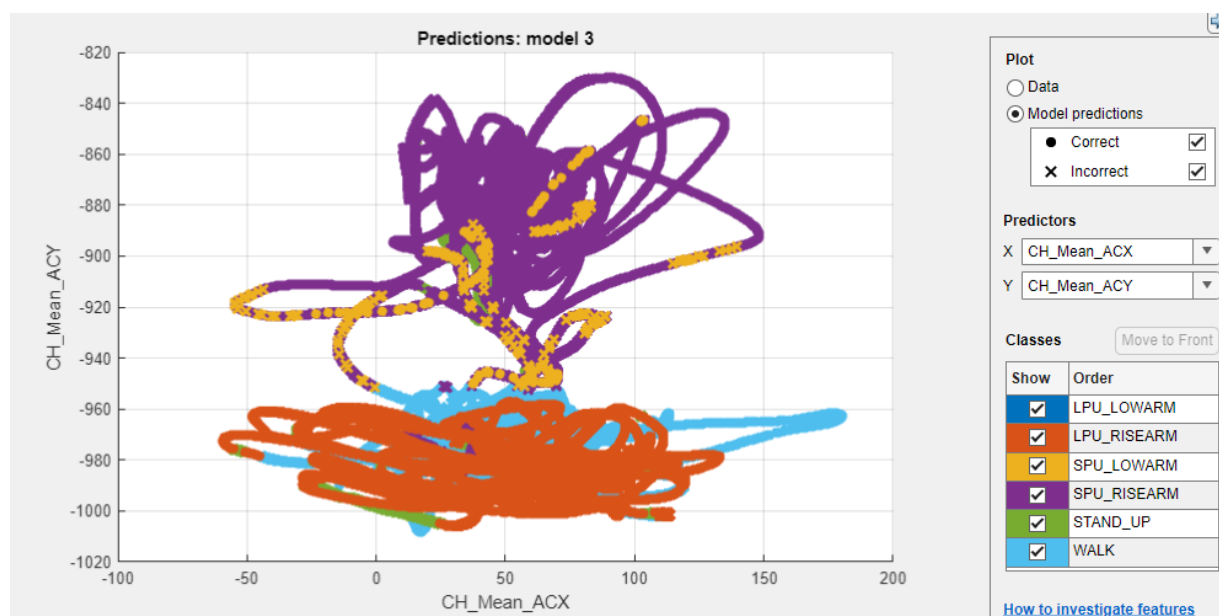


FIGURA 43 Prediction Model Coarse Tree CH

El resultado obtenido de la matriz de confusión del *Coarse Tree* de la FIGURA 44, muestra una tabla que compara las clasificaciones realizadas por el modelo con las etiquetas que se van a predecir y ayuda a identificar las áreas en las que el clasificador tuvo un desempeño eficiente o deficiente.

Las filas muestran la clase verdadera y las columnas muestran la clase pronosticada. Las celdas diagonales muestran dónde coinciden la clase verdadera y la clase predicha. Las celdas diagonales azules indican que el clasificador ha realizado la predicción correctamente de las

observaciones de esta clase verdadera. Como el algoritmo realizó un mal desempeño lo muestra en las celdas que están fuera de la diagonal la cual se observan porcentajes altos y son de color naranja. Cuanto mayor es el porcentaje, más oscuro será el tono del color de la celda. En estas celdas naranjas, la clase verdadera y la clase predicha no coinciden, por lo tanto, los puntos de datos están mal clasificados. Por lo que se puede decir que el porcentaje de efectividad de la *Coarse Tree* es bajo.

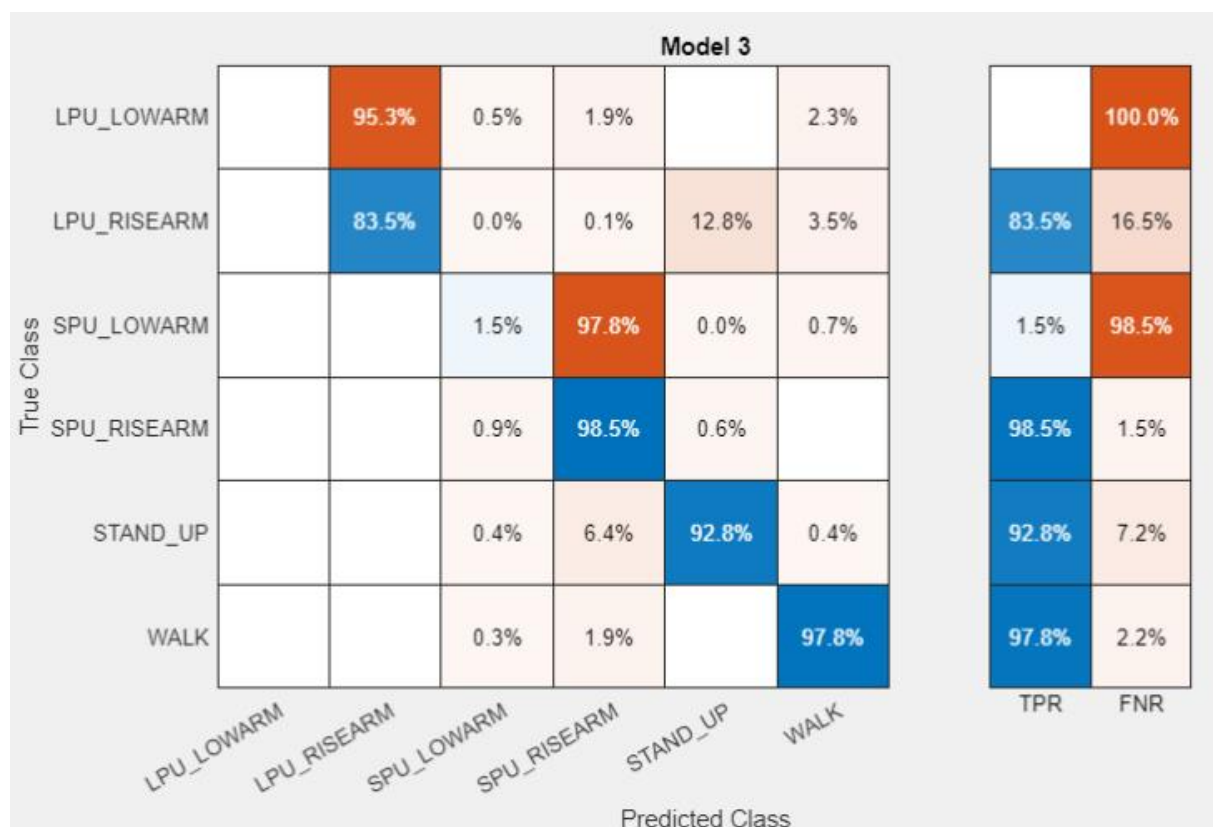


FIGURA 44 Confusion Matrix Coarse Tree CH

En la curva de ROC del *Coarse Tree* de la FIGURA 45, se observa que el porcentaje obtenido se encuentra por los valores mínimos de 0.8781, esto nos indica que el modelo entrenado para el algoritmo es de precisión baja y no permite detectar una gran cantidad de características para el entrenamiento del algoritmo, por lo tanto, no predice todas las actividades que realiza la persona mientras lleva a cabo la recolección de café.

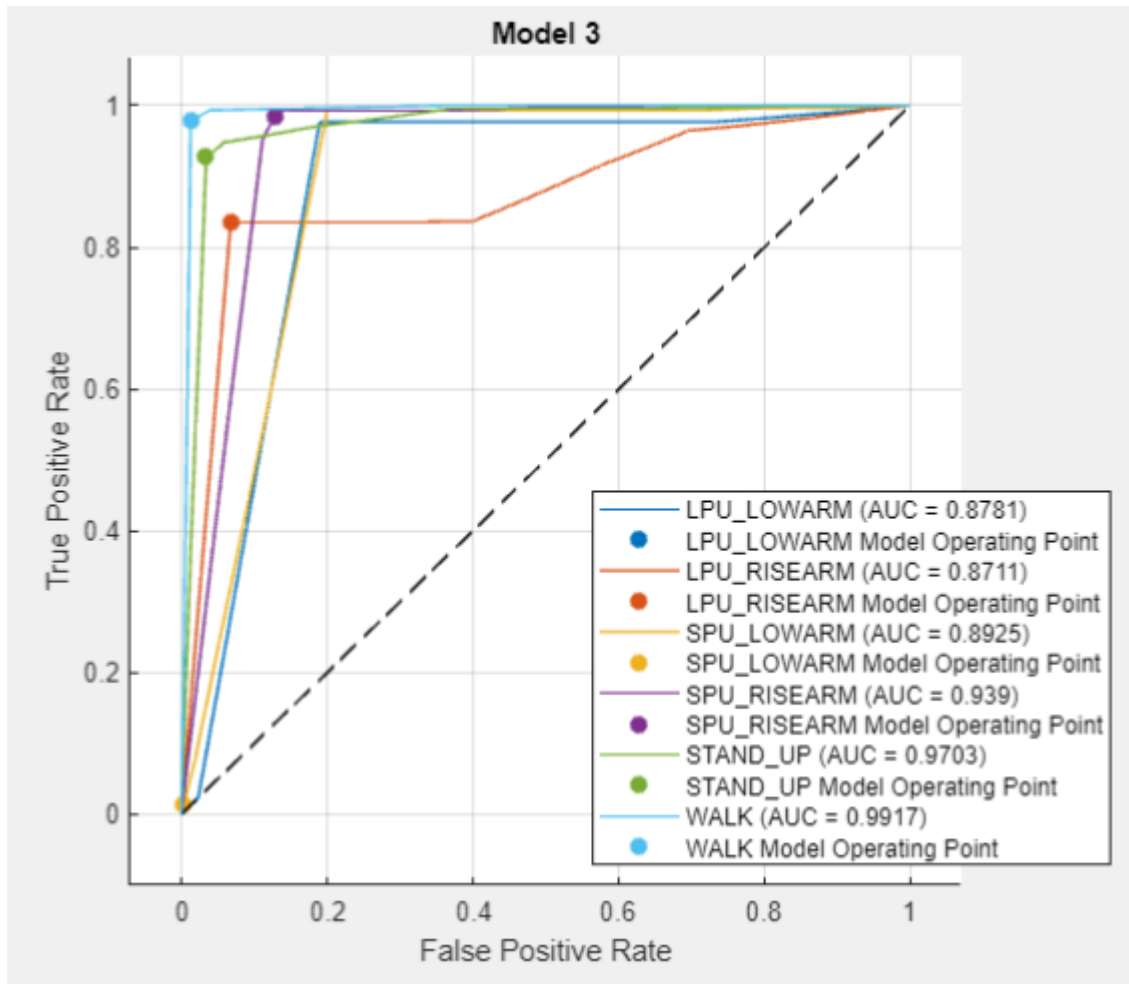


FIGURA 45 ROC Coarse Tree CH

El *Coarse Tree* de la FIGURA 46, predice las clasificaciones basadas en la característica estadística “*Peak2Peak*” comparada con el valor del giroscopio en el eje Z. Para iniciar con la predicción, se genera en el nodo superior, representado por un triángulo (Δ), la primera decisión es si el valor de “*Peak2Peak*” es menor que 8.4982, si se cumple se sigue la rama izquierda para que el árbol realice una nueva comparación. Sin embargo, si el valor de “*Peak2Peak*” es mayor al valor 8.4982, entonces se pasa hacia el nodo del triángulo de la parte inferior derecha. Aquí, el árbol vuelve a realizar la pregunta por una nueva característica para que el árbol clasifique la actividad a realizar de acuerdo con los datos entrenados. Este tipo de árbol se caracteriza por contar con un nivel simple, el cual lo hace fácil de interpretar.

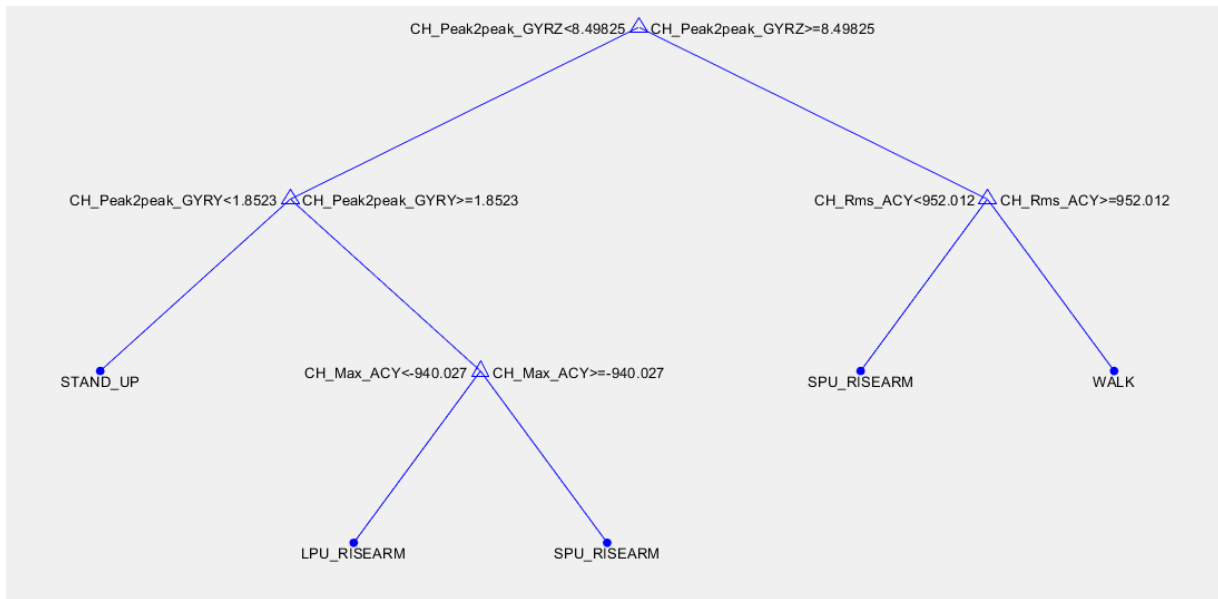


FIGURA 46 Coarse Tree CH

En el aprendizaje automático, la mejora del modelo lo hace más fácil de entender, más robusto y eficiente desde el punto de vista computacional, por lo que para obtener mejores resultados se entrena el modelo de inteligencia artificial con *Medium Tree*. En los resultados de la FIGURA 47, se logra observar un aumento de precisión en la predicción de resultado. En el resumen obtenido se puede ver las métricas del modelo para evaluar y comparar los modelos generados. Las métricas de resultados de entrenamiento calculan el conjunto de validación y muestra ahora un nuevo porcentaje obtenido con un valor del 91.4% de efectividad, así como la velocidad de predicción, el tiempo empleado para el entrenamiento y el tamaño del modelo para su exportación, por lo que se puede observar que la predicción es considerable pero su nivel de entrenamiento es medio.

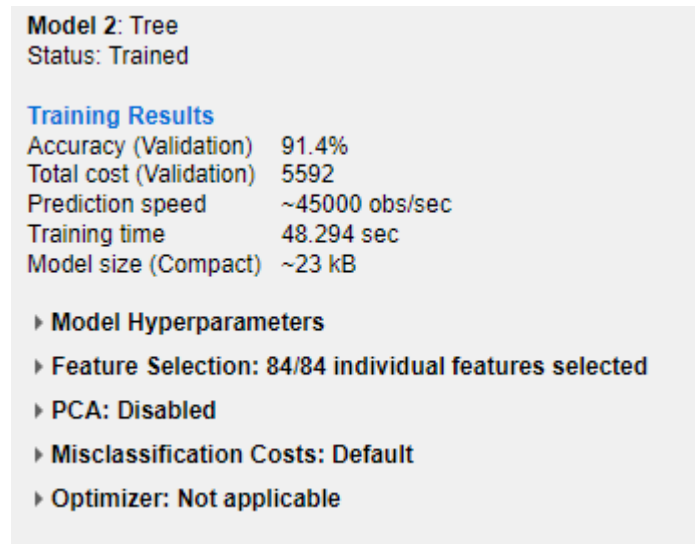


FIGURA 47 Summary Medium Tree CH

Para el caso del nuevo modelo de inteligencia artificial entrenado *Medium Tree*, en la FIGURA 48 se observan los resultados obtenidos seleccionando las características a mostrar en los ejes X e Y de los predictores, visualizando los resultados por clase mostrando u ocultando clases específicas usando las casillas de verificación de datos correctos e incorrectos, por lo que en la se figura se logra observar que indica un porcentaje medio de efectividad.

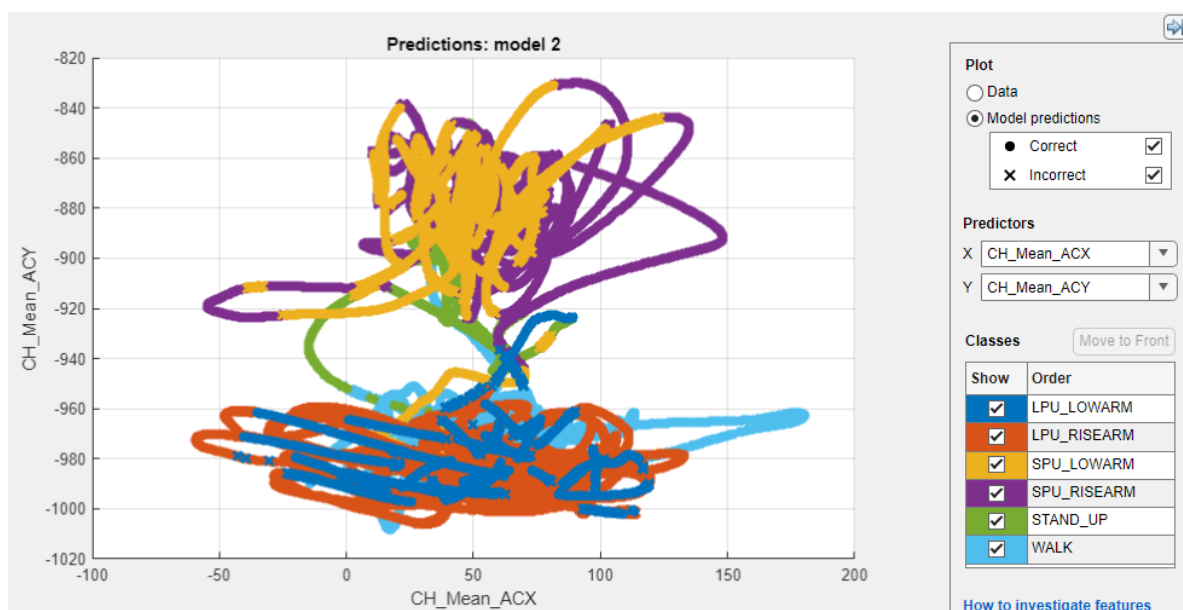


FIGURA 48 Prediction Model Medium Tree CH

En la matriz de confusión del diagrama del modelo *Medium Tree* de la FIGURA 49, se observa que la actividad “Walk” cuenta con un 99.0 %, de clasificado correctamente, por lo que el 99 % es la tasa positiva verdadera para los puntos clasificados correctamente en esta clase que se muestran en la celda azul de la columna TPR. Mientras que la actividad “Lpu_Lowarm” están mal clasificados con un 54.9% en la columna TPR. La tasa de falsos negativos para los datos clasificados incorrectamente en esta clase es del 45.1%, que se muestra en la celda naranja de la columna FNR. Los valores predictivos positivos se muestran en azul para los puntos pronosticados correctamente en cada clase, y las tasas de descubrimiento falso se muestran en naranja para los puntos pronosticados incorrectamente en cada clase. Por lo que se puede decir que el porcentaje de efectividad de la *Medium Tree* es medio.

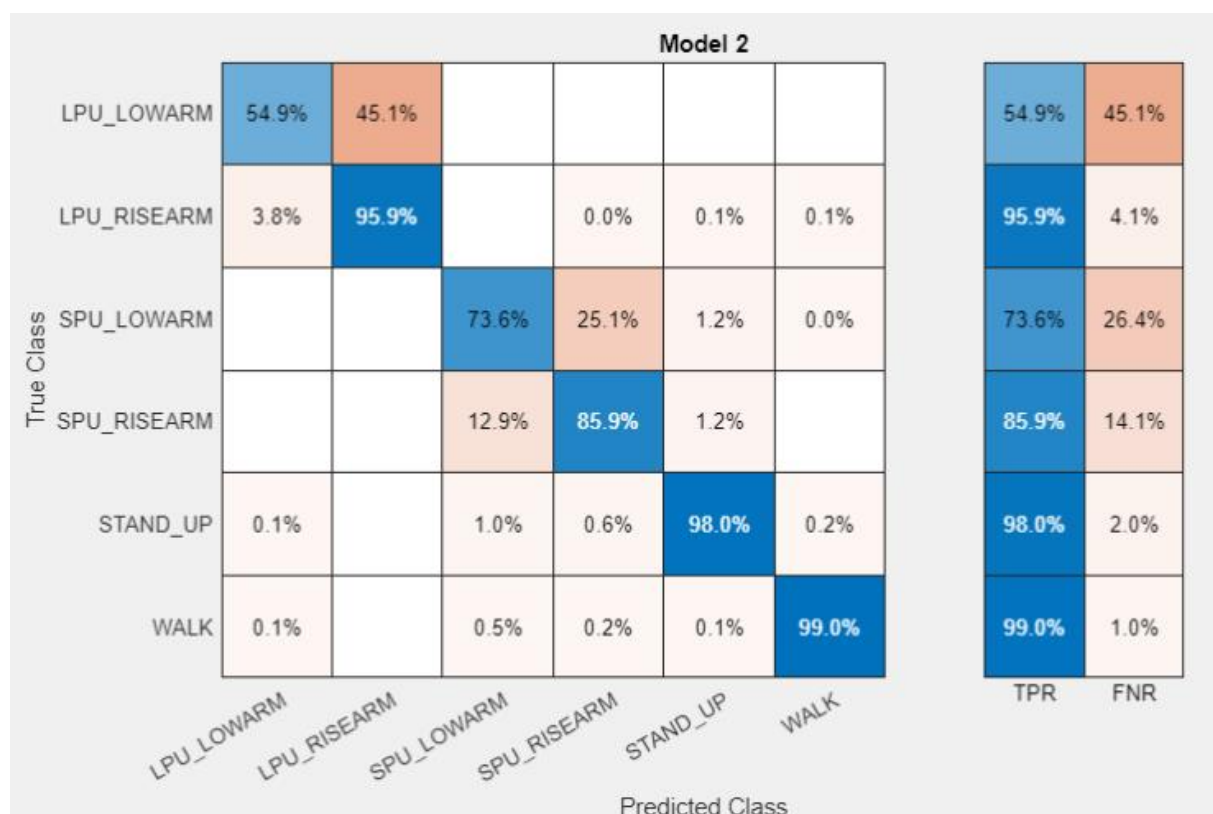


FIGURA 49 Matrix Confusion Medium Tree CH

En la curva de ROC del diagrama de *Medium Tree* en la FIGURA 50 se observa que el porcentaje mínimo es de 0.9272, esto indica que la cantidad de datos suministrada por el algoritmo es de nivel media ya que permite detectar una gran cantidad de características para el entrenamiento del algoritmo, pero los valores mostrados en la curva se muestran alejado del 1.

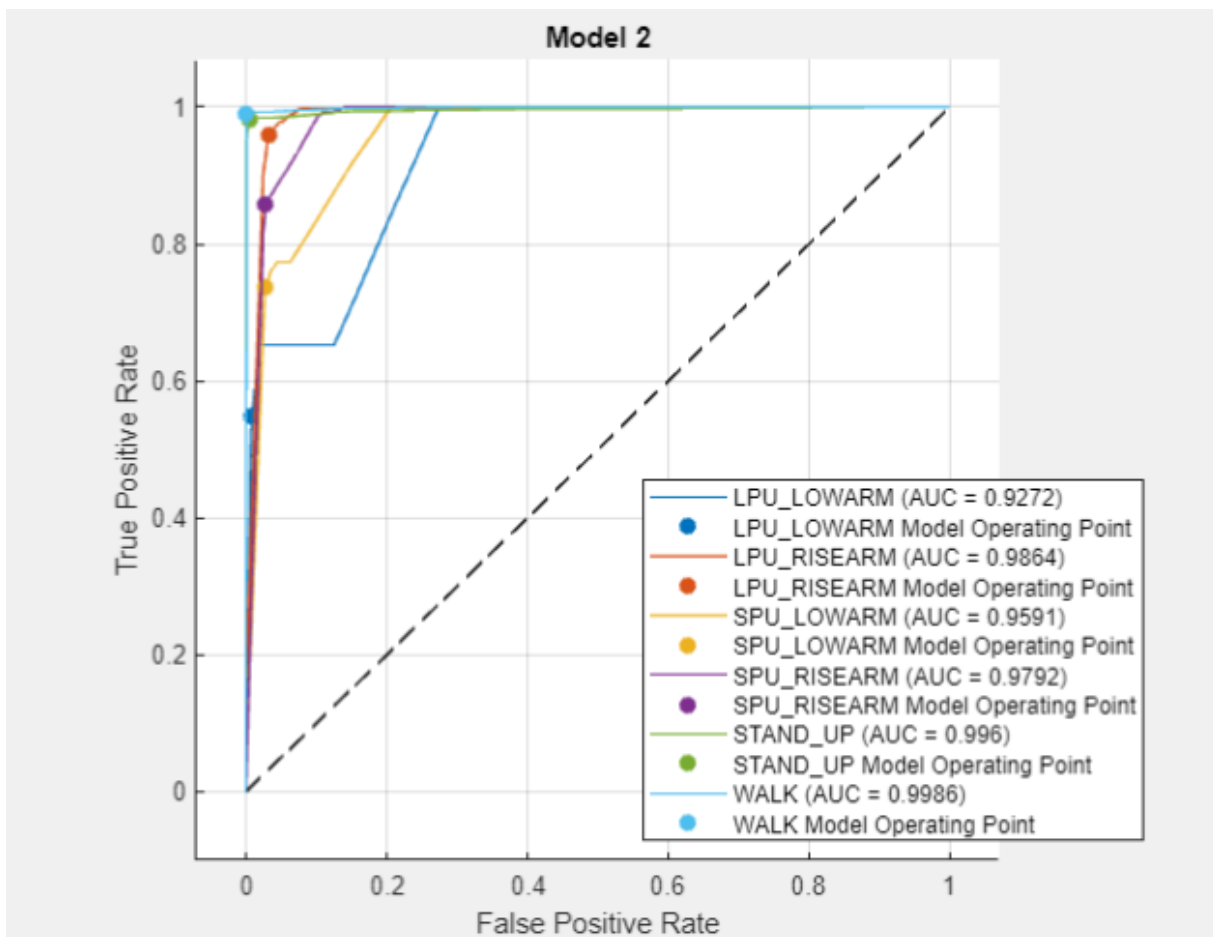


FIGURA 50 ROC Medium Tree CH

El *Medium Tree* de la FIGURA 51 predice las clasificaciones basadas en la característica estadística “Peak2Peak” comparada con el valor del giroscopio en el eje Z. Para iniciar con la predicción, se genera en el nodo superior, la primera decisión es si el valor de “Peak2Peak” es menor que 8.49825, si se cumple se sigue la rama izquierda para que el árbol realice una nueva comparación con una otra característica y de esta manera se van generando más ramas hasta llegar con la actividad a clasificar. Sin embargo, si el valor de “Peak2Peak” es mayor al valor 8.49825, entonces se pasa hacia el nodo del triángulo de la parte inferior derecha. Aquí, el árbol

vuelve a realizar la pregunta por una nueva característica para que el árbol clasifique la actividad a realizar de acuerdo con los datos entrenados. Este tipo de árbol se caracteriza por contar con un nivel medio de predicciones, el cual lo hace que se incremente los nodos con las respuestas de clasificación.



FIGURA 51 Medium Tree CH

Para mejorar las predicciones del modelo y elegir la opción más confiable para la clasificación de las actividades, para ello se entrena el modelo de inteligencia artificial con *Fine Tree* como lo muestra la FIGURA 52. En los resultados se logra observar un aumento de precisión en la predicción de resultado. En el resumen obtenido se puede ver las métricas del modelo para evaluar y comparar los modelos generados. Las métricas de resultados de entrenamiento calculan el conjunto de validación y muestra ahora un nuevo porcentaje obtenido con un valor del 97.1% de efectividad, así como la velocidad de predicción, el tiempo empleado para el entrenamiento y el tamaño del modelo para su exportación, por lo que se puede observar que la predicción es alta en comparación de los modelos entrenados.

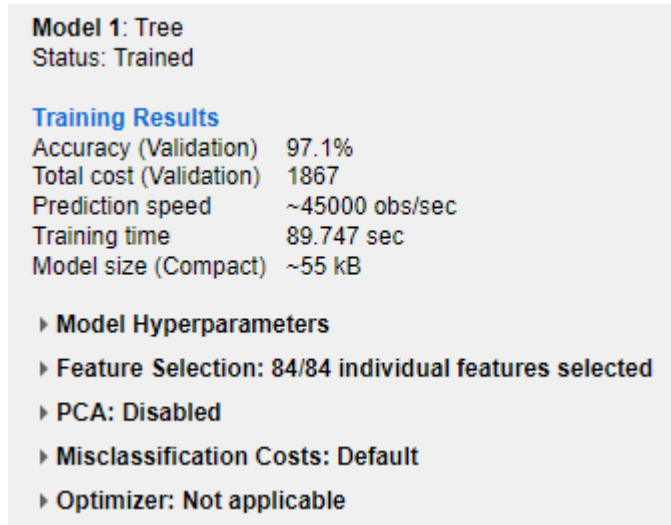


FIGURA 52 Summary Fine Tree CH

Para el caso del nuevo modelo de inteligencia artificial entrenado *Fine Tree* de la FIGURA 53 se observan los resultados obtenidos seleccionando las características a mostrar en los ejes X e Y de los predictores, visualizando los resultados por clase mostrando u ocultando clases específicas usando las casillas de verificación de datos correctos e incorrectos, por lo que en la se figura se logra observar que indica un porcentaje alto de efectividad.

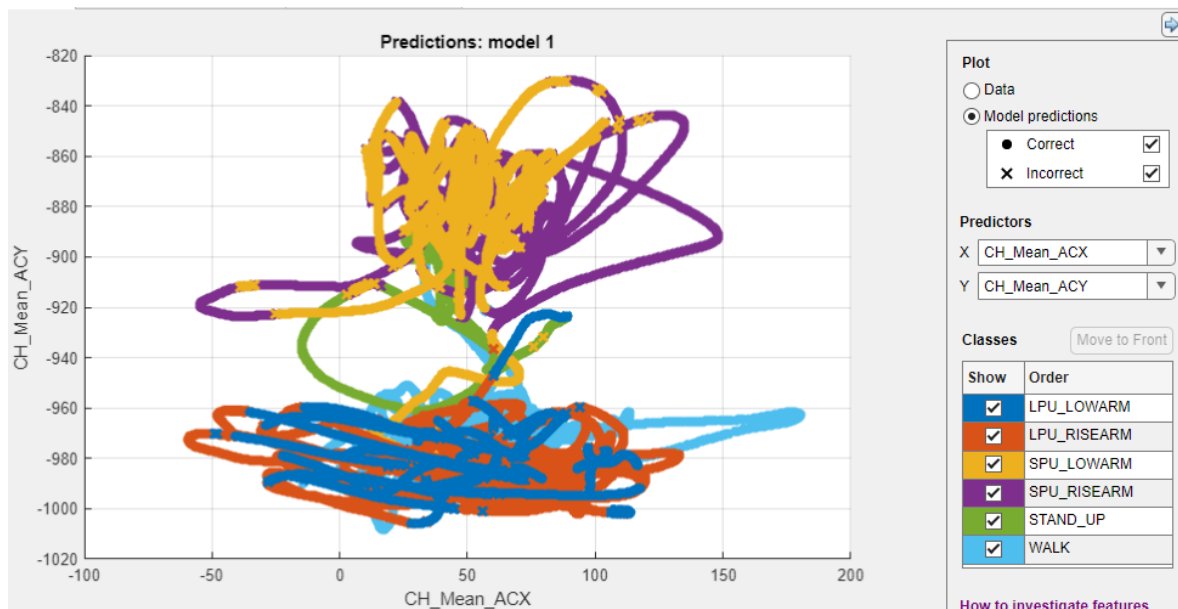


FIGURA 53 Prediction Model Fine Tree CH

En la matriz de confusión del diagrama del modelo *Fine Tree* en la FIGURA 54 se observa que la actividad “Walk” cuenta con un 99.8 %, de clasificado correctamente, por lo que el 99.8 % es la tasa positiva verdadera para los puntos clasificados correctamente en esta clase, por lo que se muestran en la celda azul de la columna TPR el porcentaje supera el 90% en las actividades a clasificar. La tasa de falsos negativos para los datos clasificados incorrectamente en esta clase es de valores mínimos como los muestra en la celda naranja de la columna FNR. Los valores predictivos positivos se muestran en azul para los puntos pronosticados correctamente en cada clase, y las tasas de descubrimiento falso se muestran en naranja para los puntos pronosticados incorrectamente en cada clase. Por lo que se puede decir que el porcentaje de efectividad del algoritmo *Fine Tree* es la más adecuada para clasificar las actividades de recolección de café en el módulo Chest.

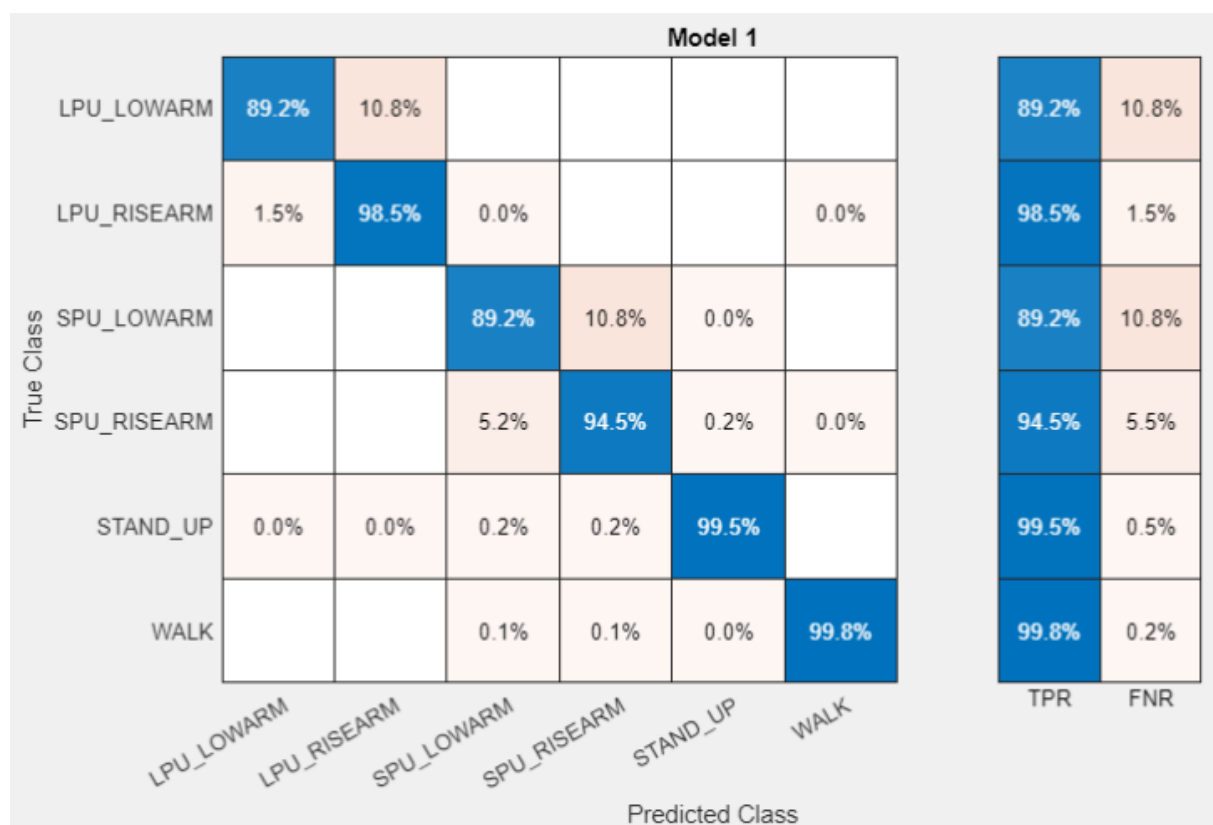


FIGURA 54 Matrix Confusion Fine Tree CH

En la curva de ROC del diagrama de *Fine Tree* en la FIGURA 55 se observa que el porcentaje mínimo es de 0.9854, esto indica que la cantidad de datos suministrada por el algoritmo es de un gran nivel alto ya que permite detectar una gran cantidad de características para el entrenamiento del algoritmo y todas las actividades clasificadas se encuentran cerca de en los valores de verdaderos positivos.

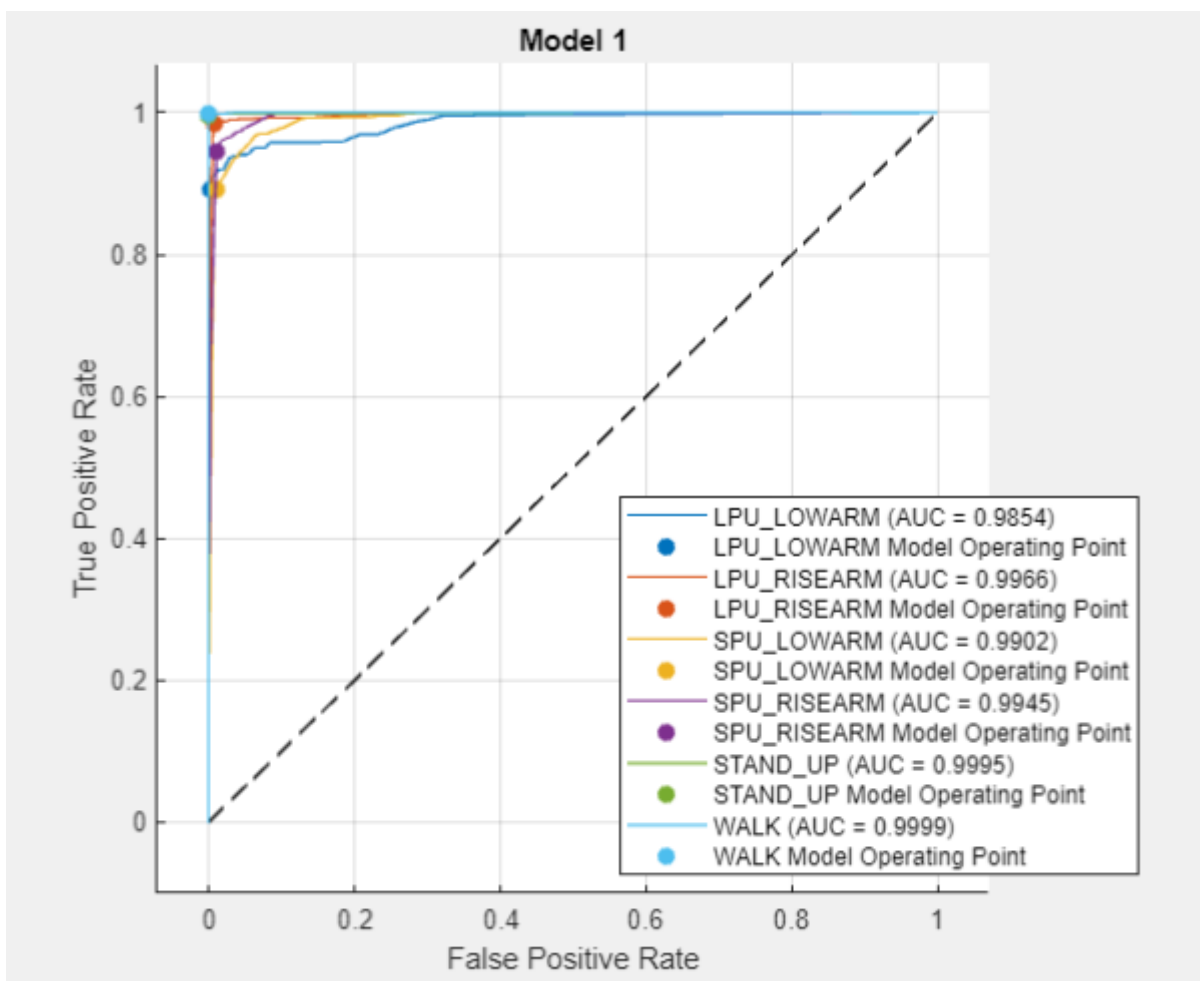


FIGURA 55 ROC Fine Tree CH

El *Fine Tree* de la FIGURA 56 redice las clasificaciones basadas en la característica estadística “Peak2Peak” comparada con el valor del giroscopio en el eje Z. Para iniciar con la predicción, se genera en el nodo superior, la primera decisión es si el valor de “Peak2Peak” es menor que 8.49825, si se cumple se sigue la rama izquierda para que el árbol realice una nueva comparación con una otra característica y de esta manera se van generando una gran cantidad de ramas de tal

para su exportación, por lo que se puede observar que sus características son rápidas pero su nivel de entrenamiento es bajo.

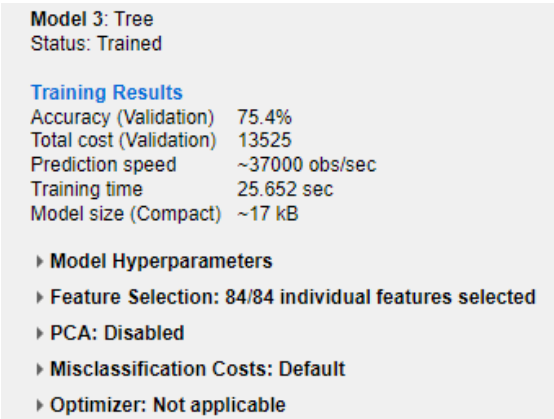


FIGURA 57 Summary Coarse Tree LA

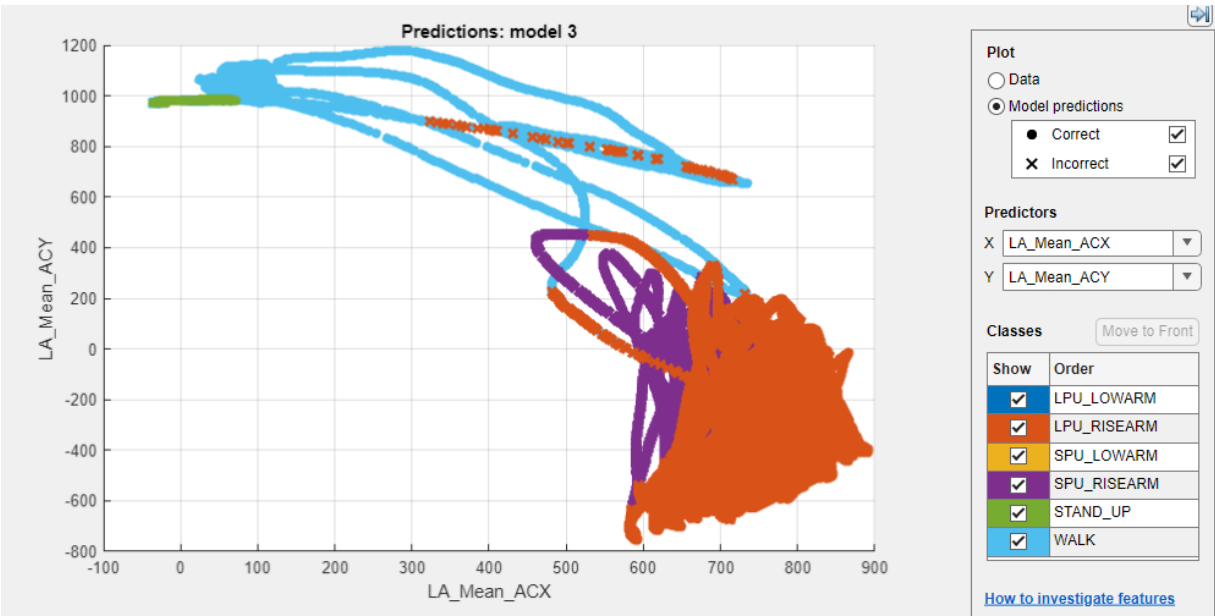


FIGURA 58 Prediction Model Coarse Tree LA

Para el caso del modelo de inteligencia artificial entrenado *Corse Tree* en la FIGURA 58 se observan los resultados obtenidos seleccionando las características a mostrar en los ejes X e Y de los predictores, visualizando los resultados por clase mostrando u ocultando clases

específicas usando las casillas de verificación de datos correctos e incorrectos, por lo que en la se figura se logra observar que indica un porcentaje bajo de efectividad.

El resultado de la matriz de confusión del *Coarse Tree* se observa en la FIGURA 59 una tabla que compara las clasificaciones del modelo con las etiquetas a predecir. Ayuda a identificar las áreas de desempeño eficiente o deficiente del clasificador. Las filas muestran la clase verdadera y las columnas muestran la clase pronosticada. Las celdas diagonales representan las coincidencias entre la clase verdadera y la clase predicha. Las celdas diagonales azules indican predicciones correctas, mientras que las celdas naranjas fuera de la diagonal muestran malas predicciones, que son las que más se observan en la matriz. Estas celdas naranjas tienen porcentajes altos y su tono de color indica la gravedad del error. Por lo que la actividad “*Lpu Lowarm*” muestra una predicción incorrecta con alto porcentaje del 98.8% En general, el desempeño de la *Coarse Tree* es bajo, ya que muestra un alto porcentaje de malas clasificaciones.

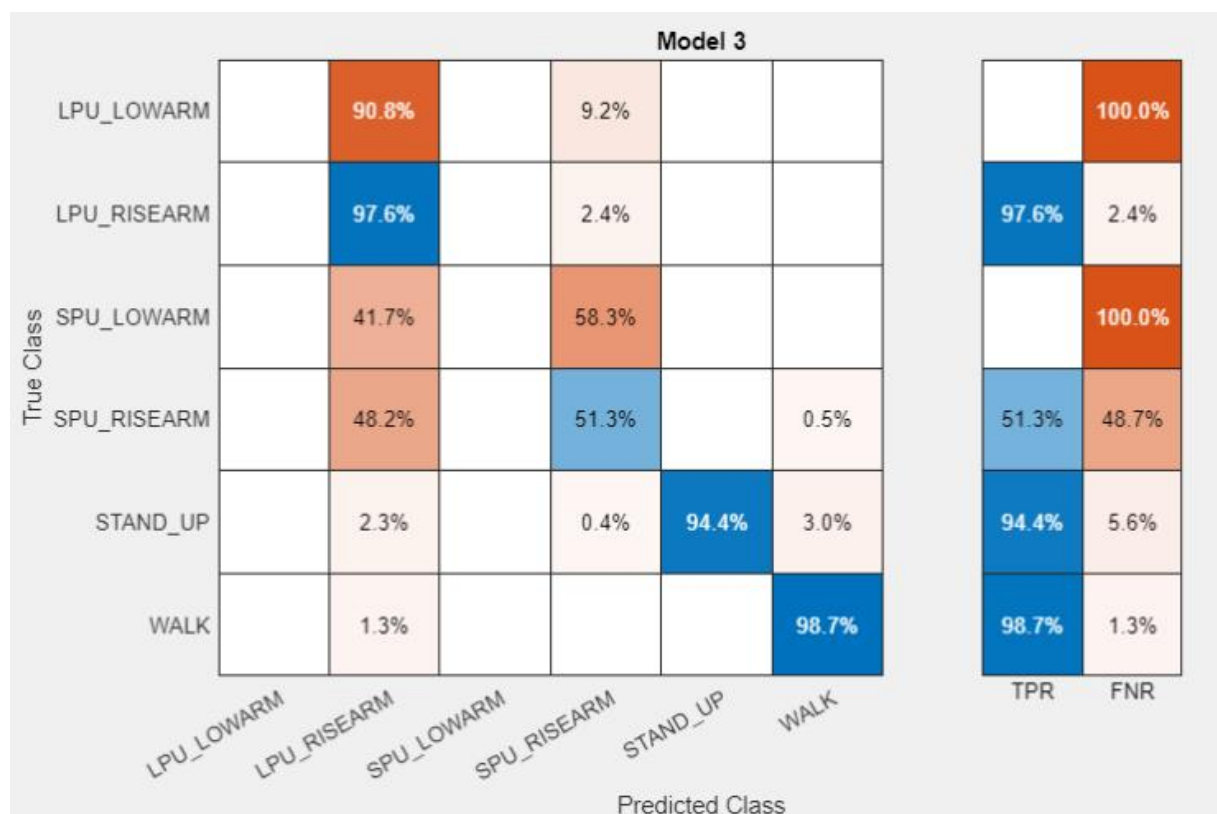


FIGURA 59 Matrix Confusion Coarse Tree LA

El diagrama de ROC del modelo *Coarse Tree* de la FIGURA 60, muestra que el porcentaje mínimo es de 0.8185. Esto implica que el algoritmo ha sido suministrado con datos de alta calidad, lo que le permite detectar numerosas características relevantes durante el entrenamiento. Además, todas las actividades clasificadas se encuentran alejadas de los valores de verdaderos positivos. En resumen, el resultado obtenido demuestra que el modelo *Coarse Tree* tiene un desempeño bajo al lograr un mínimo nivel de precisión y una incapacidad para identificar correctamente las actividades a clasificar.

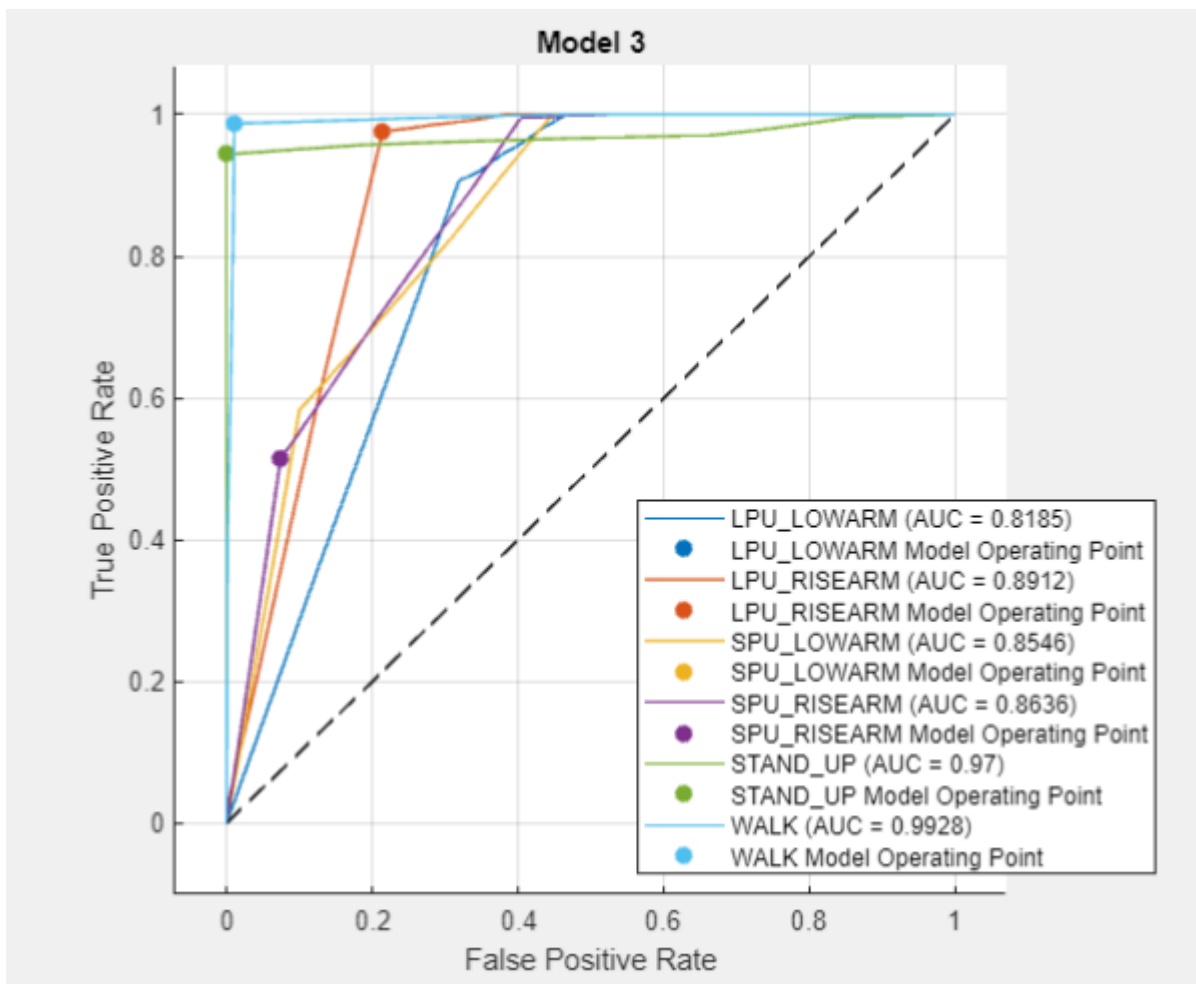


FIGURA 60 ROC Coarse Tree LA

El *Coarse Tree* de la FIGURA 61 utiliza la característica estadística “Peak2Peak” y el valor del giroscopio en el eje Y para predecir clasificaciones. Comienza en el nodo superior, representado por un triángulo (Δ), donde se toma la primera decisión basada en si el valor de “Peak2Peak” es menor que 14.7435. Si se cumple esta condición, se sigue por la rama izquierda y el árbol realiza una nueva comparación. Por otro lado, si el valor de “Peak2Peak” es mayor que 14.7435, se pasa al nodo del triángulo en la parte inferior derecha. En este nodo, el árbol plantea una nueva pregunta relacionada con otra característica para clasificar la actividad según los datos de entrenamiento. Este tipo de árbol se caracteriza por tener una estructura simple, lo que facilita su interpretación, considerando un nivel de clasificación bajo.

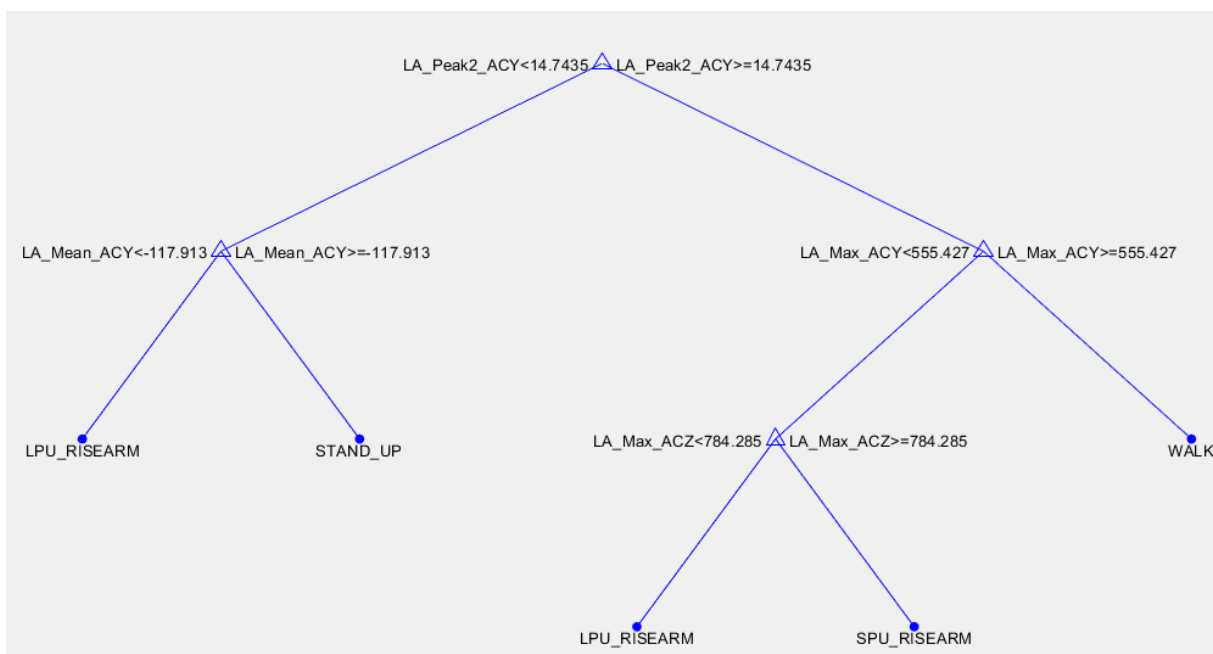


FIGURA 61 Corase Tree LA

Con el objetivo de mejorar las predicciones del modelo y seleccionar la opción más confiable para la clasificación de actividades, se utiliza el método de entrenamiento *Medium Tree* en la FIGURA 62 para el modelo de inteligencia artificial. Como resultado, se observa un aumento en la precisión de las predicciones. El resumen obtenido muestra las métricas del modelo, las cuales se utilizan para evaluar y comparar los modelos generados. Las métricas de entrenamiento muestran los resultados del conjunto de validación, con una nueva tasa de

efectividad del 87.3%. Además, se consideran la velocidad de predicción, el tiempo empleado en el entrenamiento y el tamaño del modelo para su exportación. Estos resultados indican que la precisión de la predicción es de nivel medio en comparación con el modelo previamente entrenado.

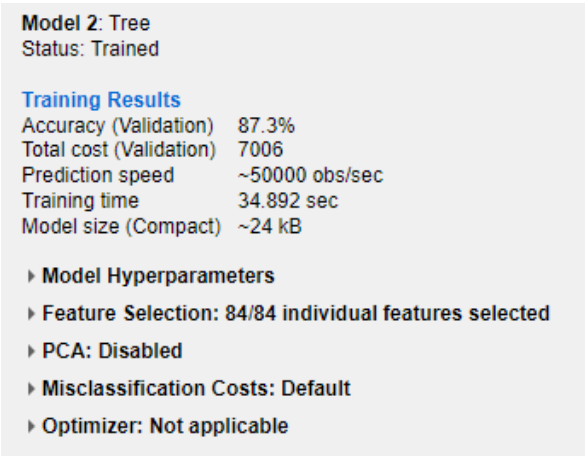


FIGURA 62 Summary Medium Tree LA

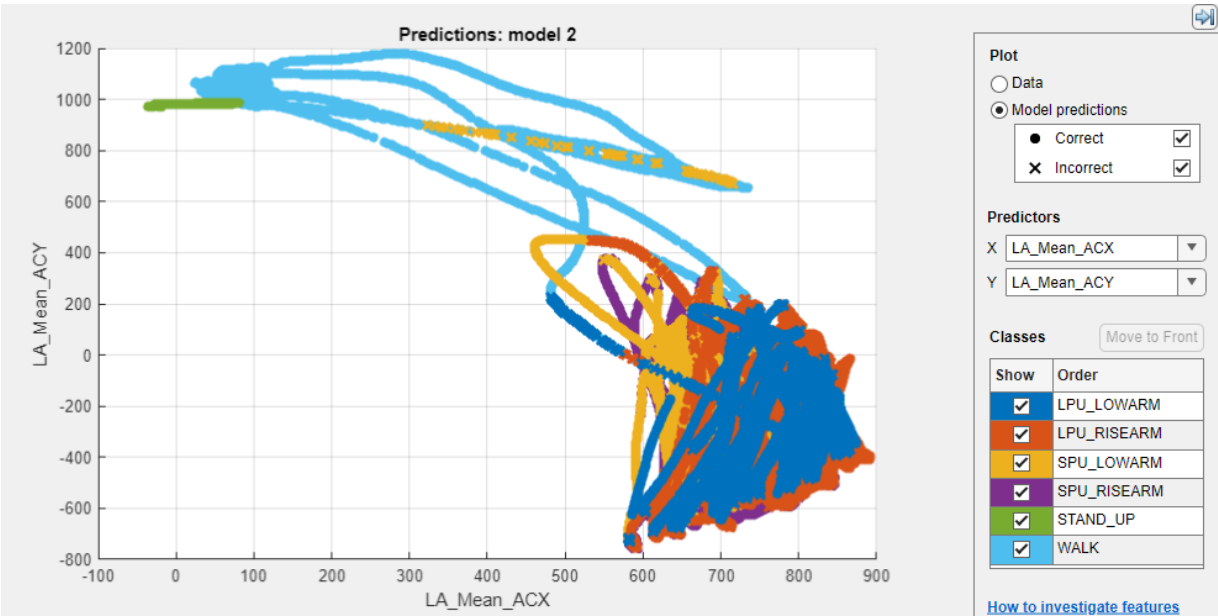


FIGURA 63 Prediction Model Medium Tree LA

En el caso del reciente modelo de inteligencia artificial entrenado *Medium Tree* en la FIGURA 63 se examinan los resultados obtenidos al seleccionar las características para mostrar en los

ejes X e Y de los predictores. Esto permite visualizar los resultados por clase, con la opción de mostrar u ocultar clases específicas utilizando casillas de verificación para datos correctos e incorrectos. En consecuencia, al observar la figura correspondiente, se puede apreciar que se obtiene un porcentaje de efectividad con nivel medio.

Al analizar la matriz de confusión del diagrama del modelo *Medium Tree* en la FIGURA 64 se puede observar que la actividad "Walk" cuenta con una clasificación correcta del 98.7%. Esto significa que el 98.7% corresponde a la tasa de verdaderos positivos, que son los puntos correctamente clasificados en esta clase, y se muestran en la celda azul de la columna TPR. Por otro lado, la actividad "Spu_Lowarm" presenta una clasificación con un mínimo porcentaje del 68.7% en la columna TPR. La tasa de falsos negativos para los datos clasificados incorrectamente se muestra en la celda naranja de la columna FNR. Los valores predictivos positivos se representan en azul para los puntos pronosticados correctamente en cada clase, mientras que las tasas de descubrimiento falso se muestran en naranja para los puntos pronosticados incorrectamente en cada clase. En resumen, se puede concluir que el modelo Medium Tree tiene una efectividad con un nivel medio.

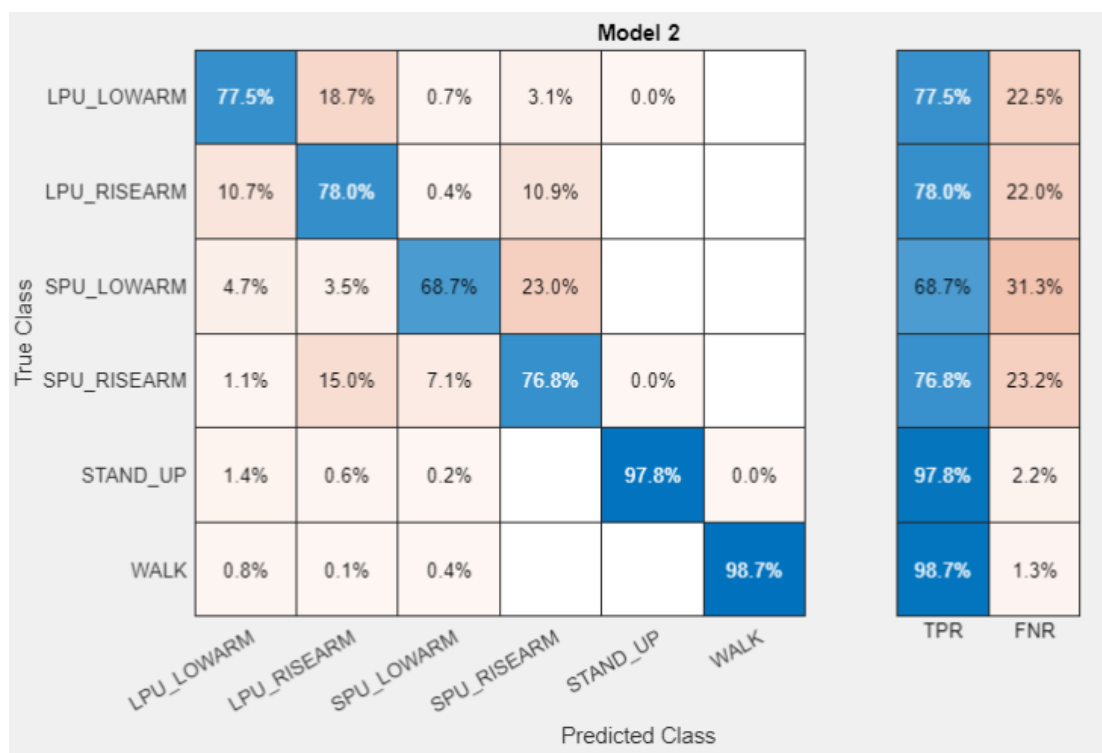


FIGURA 64 Matrix Confusion Medium Tree LA

En el diagrama de *Medium Tree* de la FIGURA 65 se puede apreciar la curva de ROC, donde se observa que el valor mínimo de porcentaje es de 0.9407. Esto sugiere que la cantidad de datos proporcionados por el algoritmo se sitúa en un nivel medio, lo que permite detectar solo ciertas características para el entrenamiento del algoritmo.

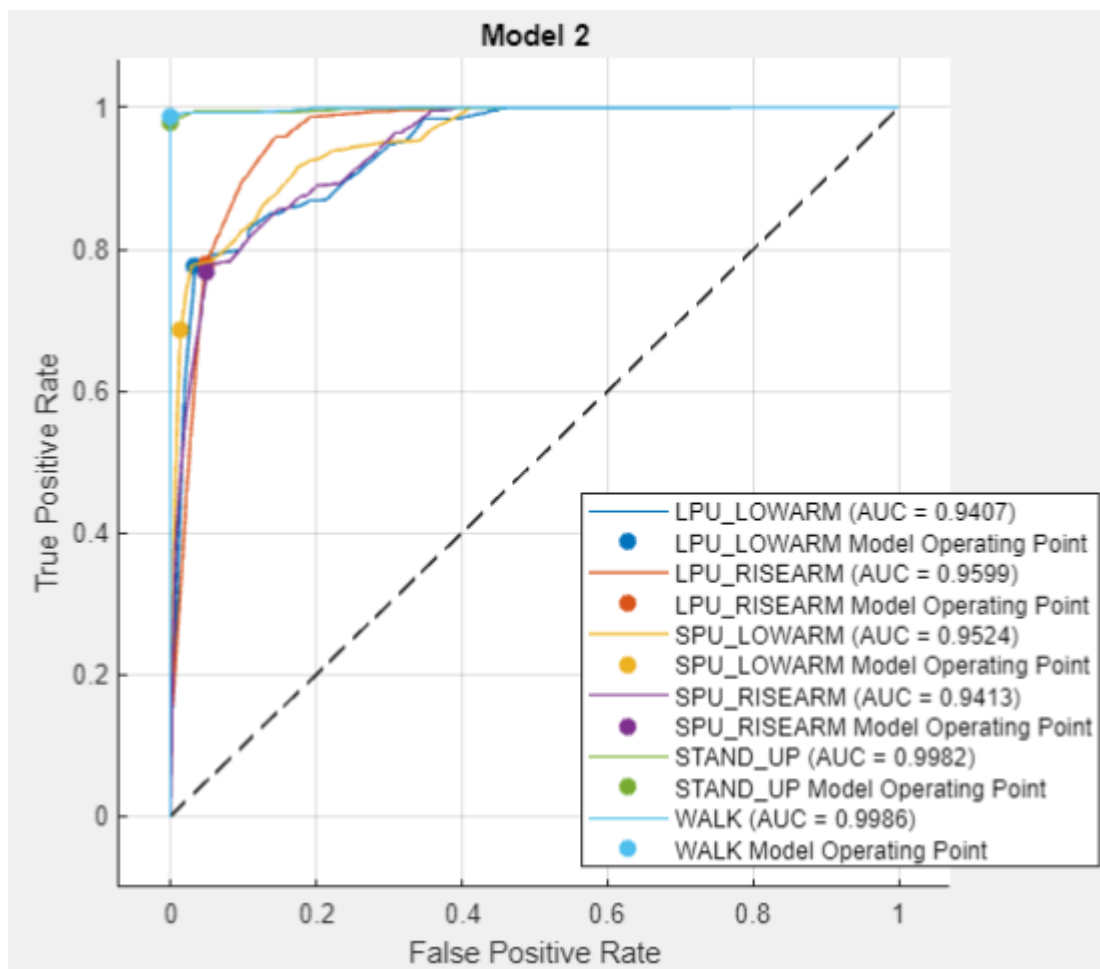


FIGURA 65 ROC Medium Tree LA

El modelo Medium Tree de la FIGURA 66 realiza predicciones basadas en la característica estadística “Peak2Peak” en comparación con el valor del acelerómetro en el eje Y. Para comenzar la predicción, se inicia en el nodo superior y se toma la primera decisión según si el valor de “Peak2Peak” es menor que 14.7435. Si se cumple esta condición, se sigue por la rama izquierda del árbol y se realiza una nueva comparación con otra característica, generando así más ramas hasta llegar a la actividad a clasificar. Sin embargo, si el valor de “Peak2Peak” es

[illegible]

FIGURA 66 Medium Tree LA

Con el objetivo de mejorar las predicciones del modelo y seleccionar la opción más confiable para la clasificación de actividades, se entrena el modelo de inteligencia artificial utilizando *Fine Tree* como se observa en la FIGURA 67, Como resultado, se observa un aumento en la precisión de las predicciones. En el resumen obtenido, se presentan las métricas del modelo para evaluar y comparar los modelos generados. Las métricas de resultados de entrenamiento calculan el conjunto de validación y muestran un nuevo porcentaje de efectividad de 95.2%, junto con la velocidad de predicción, el tiempo empleado en el entrenamiento y el tamaño del modelo para su exportación. Esto demuestra que la efectividad de la predicción es alta superando así los modelos previamente entrenados.

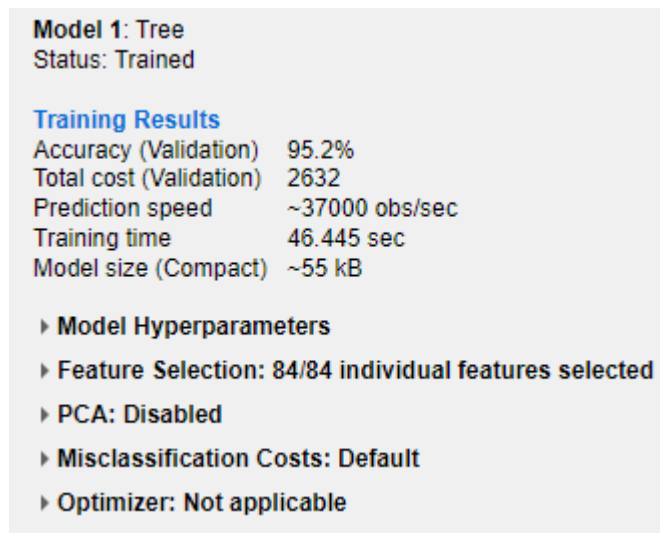


FIGURA 67 Summary Fine Tree LA

En relación con el nuevo modelo de inteligencia artificial entrenado llamado *Fine Tree* en la FIGURA 68 se analizan los resultados al elegir características específicas para mostrar en los ejes X e Y de los predictores. Esto facilita la visualización de los resultados por cada clase, dándonos la opción de mostrar u ocultar clases específicas mediante casillas de verificación para datos correctos e incorrectos. Al examinar la figura correspondiente, se puede observar que se obtiene un nivel alto de efectividad en términos de porcentaje.

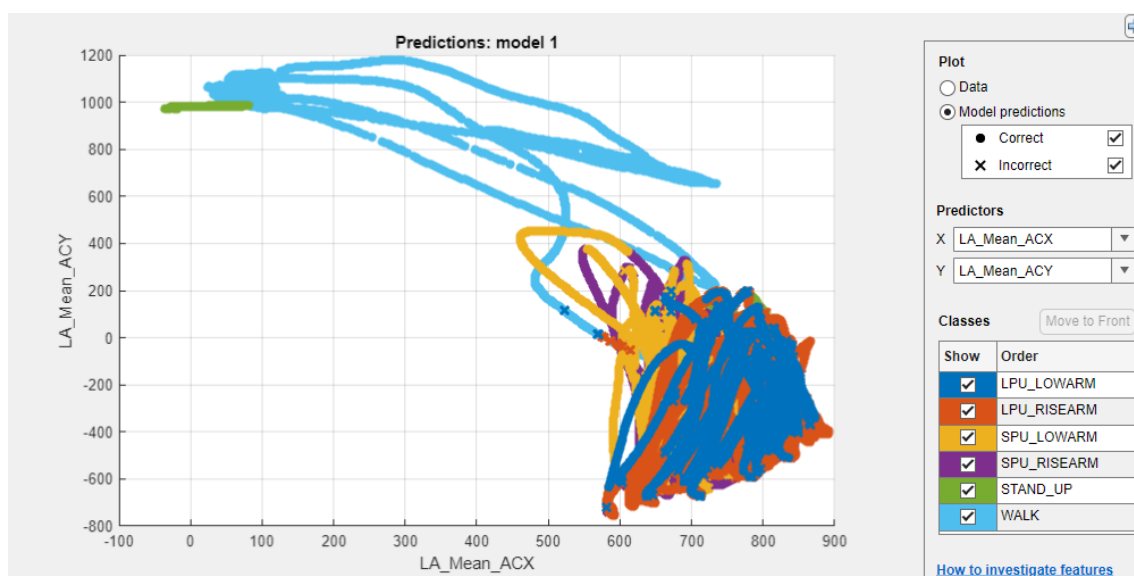


FIGURA 68 Prediction Model Fine Tree LA

La representación gráfica de la matriz de confusión brinda información sobre el rendimiento del clasificador en el modelo *Fine Tree* en la FIGURA 69 se puede observar que la actividad "Walk" y "Stand Up" cuentan con una precisión de clasificación del 99.9%. Esto significa que el 99.9% de los puntos en esta clase están correctamente clasificados, lo cual se refleja en la celda azul de la columna TPR. El porcentaje de clasificación correcta en las actividades a clasificar supera el 90%. Además, la tasa de falsos negativos, es decir, los datos clasificados incorrectamente en esta clase, es mínima, como se muestra en la celda naranja de la columna FNR. En resumen, el algoritmo *Fine Tree* demuestra una alta eficacia para clasificar las actividades de recolección de café en el módulo *Left Arm*.

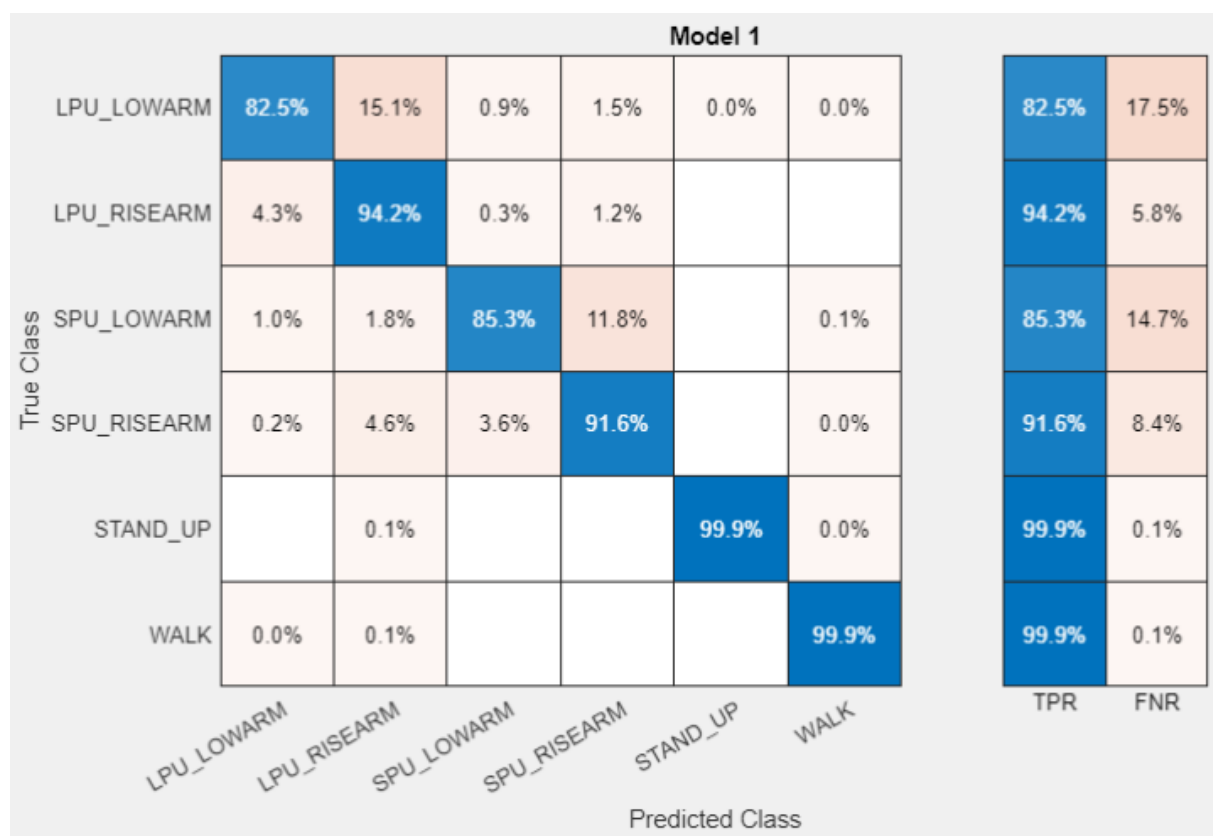


FIGURA 69 Matrix Confusion Fine Tree LA

La curva de ROC proporciona la información sobre el rendimiento del algoritmo *Fine Tree* como se muestra en la FIGURA 70, clasificándola con un buen nivel. Sin embargo, el porcentaje mostrado de la curva no es una medida representativa de la efectividad del algoritmo. Esta efectividad se determina mediante la relación entre los porcentajes de valores positivos y negativos, lo que permite predecir si la curva se inclina hacia el punto exacto de clasificación. Los valores se acercan a 1 indicando una clasificación positiva y se muestran alejados del valor de 0 indicando una clasificación negativa. La curva de ROC del *Fine tree* se muestra la gráfica se muestra en

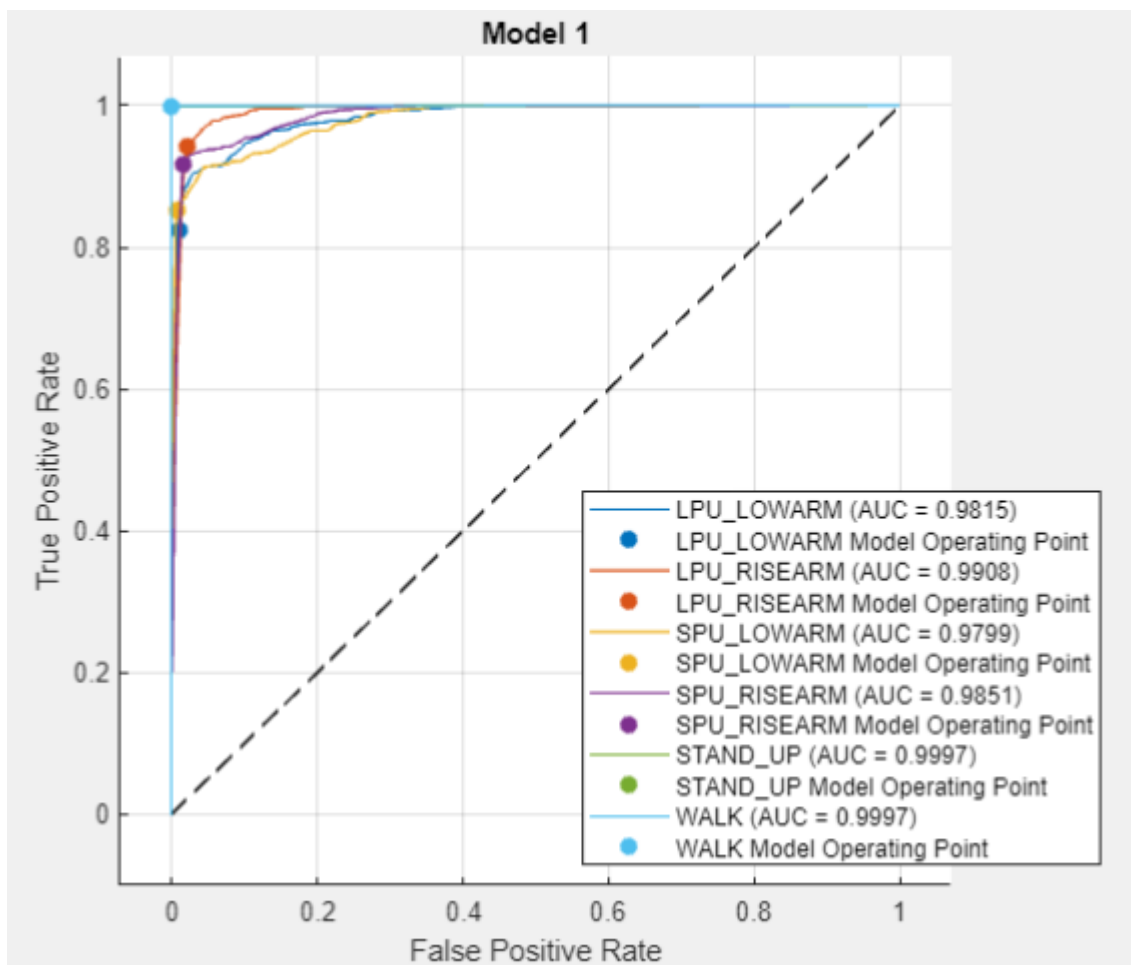
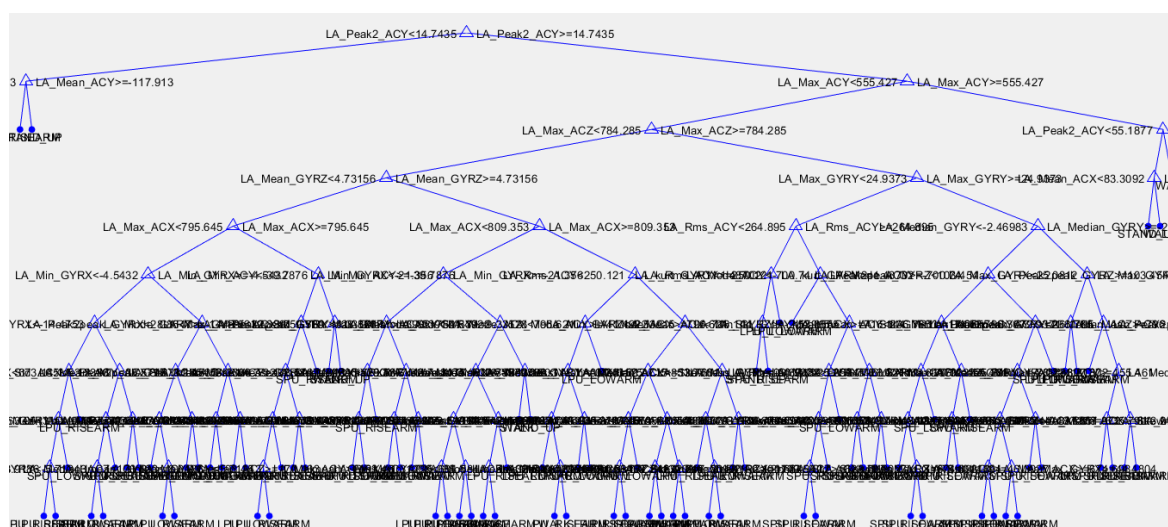


FIGURA 70 ROC Fine Tree LA

Por otro lado, si el valor de “Peak2Peak” es mayor a 14.7435, se continúa por el nodo del triángulo en la parte inferior derecha. En este caso, el árbol realiza una nueva comparación con una característica distinta y un eje diferente para clasificar la actividad según los datos de entrenamiento. Se llevan a cabo diferentes comparaciones para lograr una clasificación más precisa de las actividades. Este tipo de árbol se destaca por tener un alto nivel de predicciones, lo cual implica un mayor número de nodos con respuestas de clasificación para el módulo Left Arm.



4.3. Modulo RA

Para obtener los modelos de inteligencia artificial del módulo *Right Arm*, se realizó el entrenamiento de datos utilizando el algoritmo *Coarse Tree* de la FIGURA 72. Sin embargo, al analizar los resultados obtenidos, se observa que no se logra alcanzar una alta precisión en el entrenamiento. Esto se refleja en la baja capacidad del modelo para clasificar todas las actividades realizadas por la persona durante la recolección de café.

En el resumen de los resultados, se presentan las métricas del modelo para su evaluación y comparación con otros modelos generados. Las métricas de entrenamiento calculan el rendimiento en el conjunto de validación, donde apenas se alcanza un porcentaje de efectividad del 78.6%. Además, se proporciona información sobre la velocidad de predicción, el tiempo empleado en el entrenamiento y el tamaño del modelo para su exportación. Aunque se destaca la rapidez de las características del modelo, se evidencia que su nivel de entrenamiento es bajo

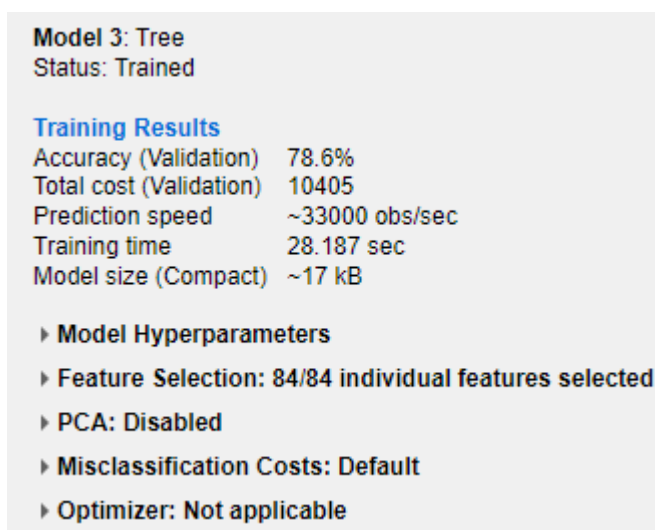


FIGURA 72 Summary Coarse Tree RA

En el caso del modelo de inteligencia artificial entrenado *Coarse Tree* en la FIGURA 73 se examinan los resultados obtenidos al seleccionar las características a mostrar en los ejes X e Y de los predictores. Esto permite visualizar los resultados por clase, con la opción de mostrar u ocultar clases específicas utilizando casillas de verificación para datos correctos e incorrectos. Al observar la figura correspondiente, se puede notar que se obtiene un porcentaje bajo de efectividad en la clasificación.

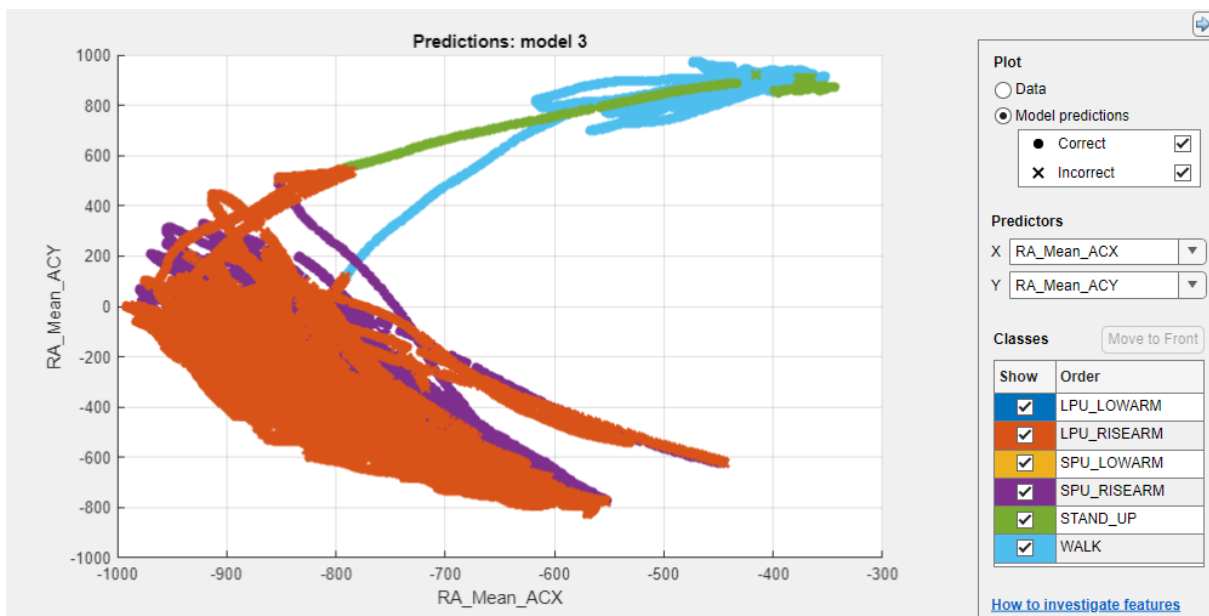


FIGURA 73 Prediction Model Coarse Tree RA

El resultado de la matriz de confusión generada por el modelo *Coarse Tree* se observa en la FIGURA 74 una tabla que permite comparar las clasificaciones del modelo con las etiquetas objetivo, por lo que se van a identificar las áreas de rendimiento efectivo o deficiente del clasificador. Las filas representan las clases reales, mientras que las columnas muestran las clases predichas. Las celdas en la diagonal principal indican las coincidencias entre la clase verdadera y la clase pronosticada, representando las predicciones correctas en color azul. Las celdas fuera de la diagonal principal, resaltadas en color naranja, indican las predicciones incorrectas y su tonalidad refleja la magnitud del error.

En particular, se observa que la actividad "Lpu Lowarm" tiene una alta tasa de predicciones incorrectas, alcanzando un 100% en los resultados. En general, el rendimiento del modelo

Coarse Tree es bajo, ya que muestra la actividad “Spu Lowarm” con un 100% de clasificación lo que indica que no hay ninguna aproximación en la columna FNR.



FIGURA 74 Matrix Confusion Coarse Tree RA

La curva de ROC del modelo *Coarse Tree* en la FIGURA 75 revela que el porcentaje más bajo obtenido es de 0.8482. Esto indica que todas las actividades clasificadas se encuentran alejadas de los valores de verdaderos positivos. En resumen, los resultados obtenidos evidencian que el modelo *Coarse Tree* presenta un rendimiento deficiente, ya que alcanza un nivel mínimo de precisión y no logra identificar correctamente las actividades a clasificar.

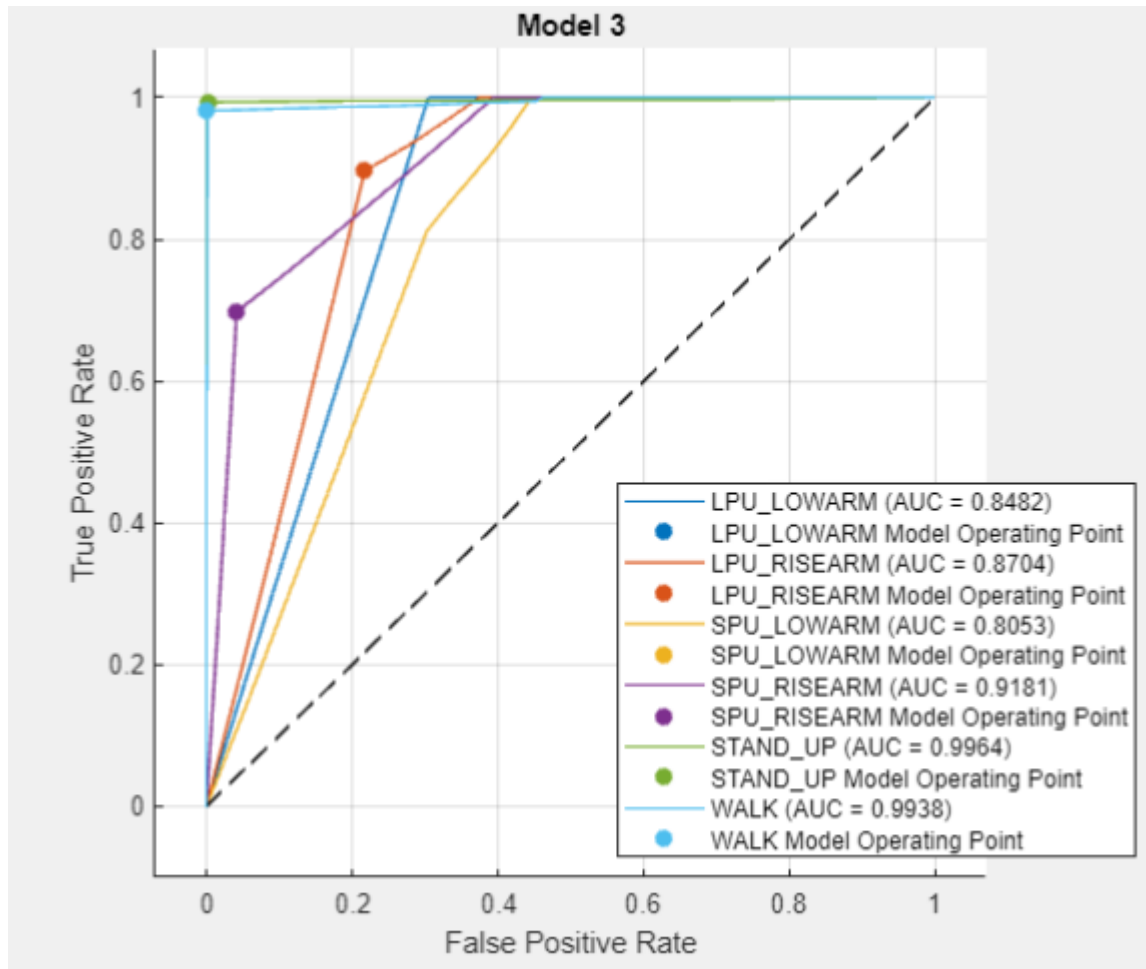


FIGURA 75 ROC Coarse Tree RA

El *Coarse Tree* de la FIGURA 76, utiliza la característica estadística “Peak2Peak” y el valor del giroscopio en el eje Z para realizar predicciones de clasificación. Comienza en un nodo superior representado por un triángulo (Δ), donde se toma la primera decisión basada en si el valor de “Peak2Peak” es menor que 4.5614. Si se cumple esta condición, se sigue por la rama izquierda del árbol y se realiza una nueva comparación. En caso contrario, si el valor de “Peak2Peak” es mayor que 4.5614, se pasa al nodo del triángulo en la parte inferior derecha. En este nodo, el árbol plantea una nueva pregunta relacionada con otra característica para clasificar la actividad según los datos de entrenamiento. Este tipo de árbol se caracteriza por su estructura simple, lo que facilita su interpretación y se considera de bajo nivel de predicción.

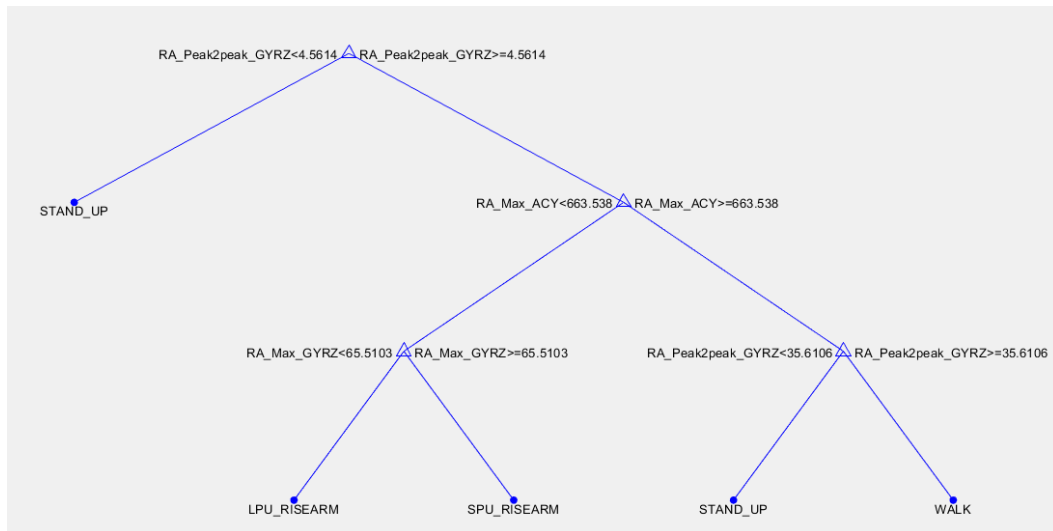


FIGURA 76 Coarse Tree RA

Con la finalidad de mejorar las predicciones del modelo y seleccionar la opción más confiable para la clasificación de actividades, se utiliza el método de entrenamiento *Medium Tree* en el modelo de inteligencia artificial. Como resultado, se observa un incremento en la precisión de las predicciones.

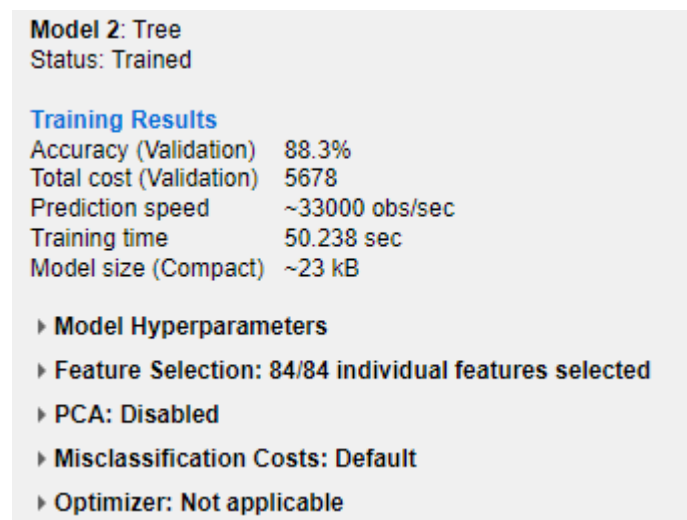


FIGURA 77 Summary Medium Tree RA

El resumen obtenido en la FIGURA 77 muestra las métricas del modelo, que se utilizan para evaluar y comparar los modelos generados. Las métricas de entrenamiento reflejan los

resultados del conjunto de validación, con una nueva tasa de efectividad del 88.3%. También se consideran la velocidad de predicción, el tiempo empleado en el entrenamiento y el tamaño del modelo para su exportación. Estos resultados indican que la precisión de la predicción se encuentra a un nivel medio en comparación con el modelo previamente entrenado.

En el caso del nuevo modelo de inteligencia artificial entrenado con *Medium Tree* lo muestra la FIGURA 78, se analizan los resultados al seleccionar las características para mostrar en los ejes X e Y de los predictores. Esto facilita la visualización de los resultados por clase y ofrece la opción de mostrar u ocultar clases específicas mediante casillas de verificación para datos correctos e incorrectos. Al examinar la figura correspondiente, se puede observar que se logra un nivel medio de efectividad.

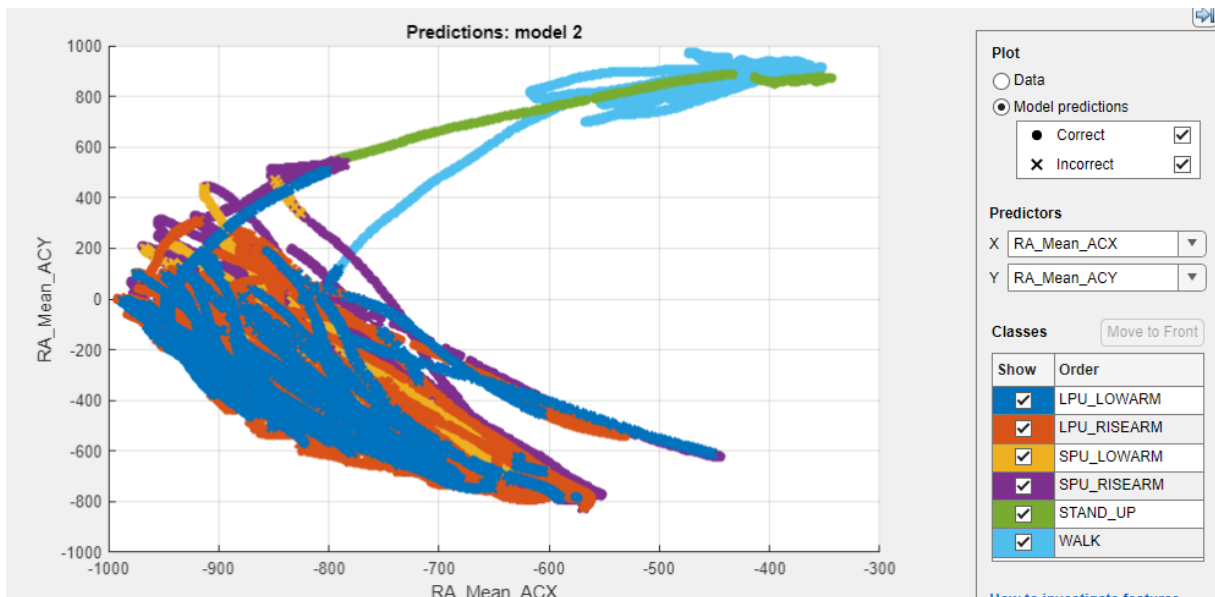


FIGURA 78 Prediction Model Medium Tree RA

Al examinar la matriz de confusión del gráfico del modelo *Medium Tree* en la FIGURA 79, se puede notar que la actividad "Stand Up" tiene una tasa de clasificación correcta del 99.6%. Esto implica que el 99.6% corresponde a la proporción de verdaderos positivos, que son los puntos correctamente clasificados en esta categoría, y se representan en la celda azul de la columna TPR. Por otro lado, la actividad "Spu_Lowarm" presenta una clasificación con una tasa mínima del 55.4% en la columna TPR. La tasa de falsos negativos para los datos clasificados incorrectamente se muestra en la celda naranja de la columna FNR. Los valores predictivos positivos se indican en azul para los puntos pronosticados correctamente en cada categoría, mientras que las tasas de descubrimiento falso se presentan en naranja para los puntos pronosticados incorrectamente en cada categoría. En resumen, se puede concluir que el modelo *Medium Tree* tiene una efectividad de nivel medio.

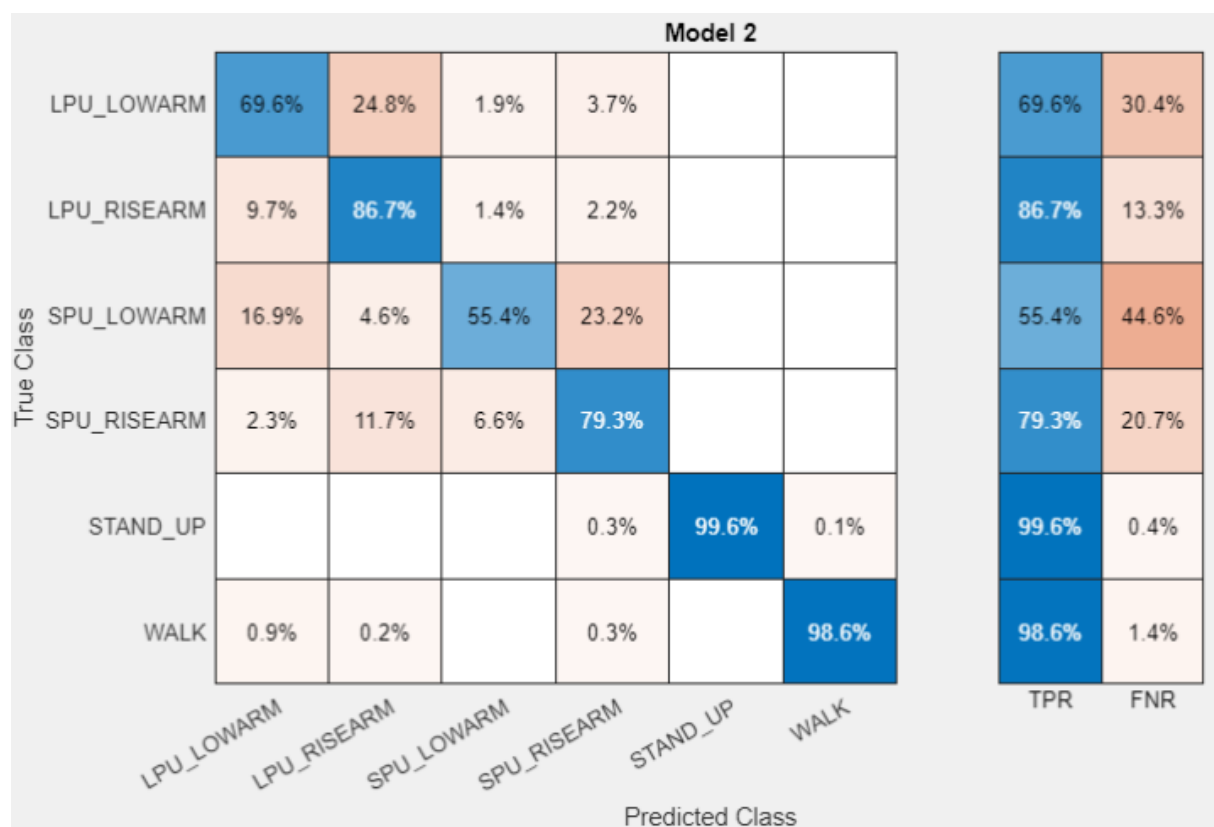


FIGURA 79 Matrix Confusion Medium Tree RA

En el diagrama del modelo *Medium Tree* como lo muestra la FIGURA 80, se puede visualizar claramente la curva de ROC, la cual muestra que el valor mínimo de porcentaje obtenido es de 0.9351. Esta evidencia nos lleva a concluir que la cantidad de datos suministrados por el algoritmo se encuentra en un rango intermedio, lo que implica que solo se pueden identificar y utilizar ciertas características para el proceso de entrenamiento del algoritmo.

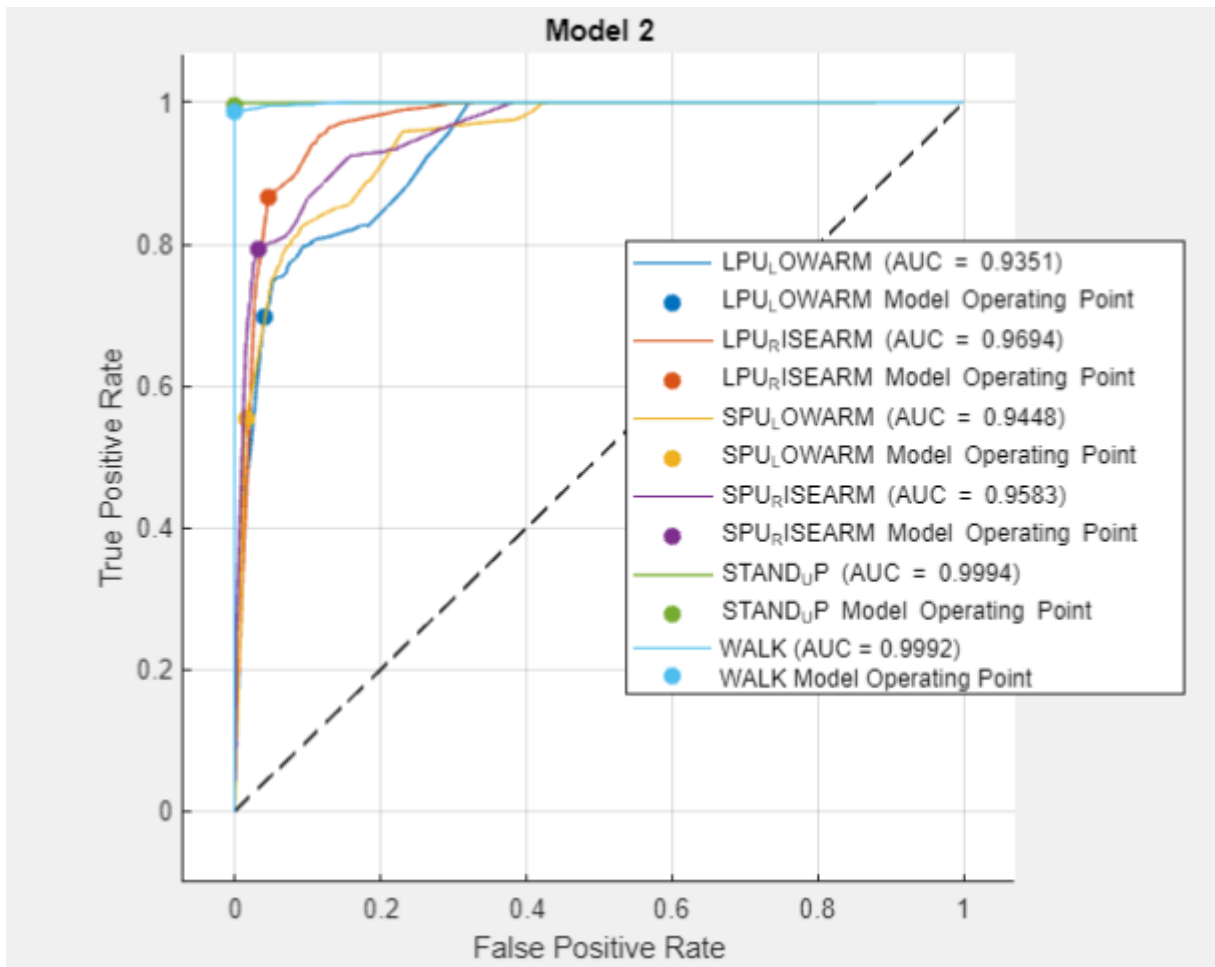


FIGURA 80 ROC Medium Tree RA

El modelo *Medium Tree* mostrado en la FIGURA 81, emplea la característica estadística “Peak2Peak” y compara su valor con el del giroscopio en el eje Z para realizar predicciones. Para iniciar la predicción, se comienza en el nodo superior y se toma la primera decisión según si el valor de “Peak2Peak” es menor a 4.5614. Si se cumple esta condición, se sigue por la rama izquierda del árbol y se realiza una comparación con otra característica, lo que genera nuevas ramas hasta llegar a la actividad a clasificar. Sin embargo, si el valor de “Peak2Peak” es mayor a 4.5614, se pasa al nodo del triángulo en la parte inferior derecha. En este nodo, el árbol realiza otra pregunta basada en una nueva característica y eje diferente para clasificar la actividad en base a los datos de entrenamiento. Este tipo de árbol se caracteriza por tener un nivel intermedio de predicciones, lo cual implica que hay un mayor número de nodos con respuestas de clasificación.

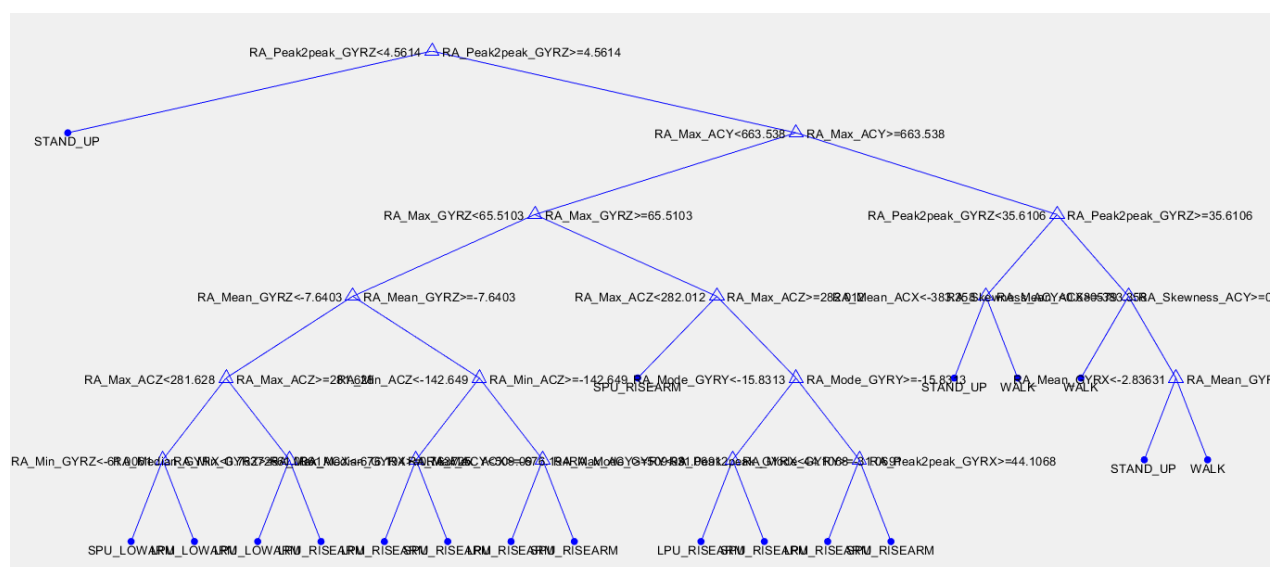


FIGURA 81 Medium Tree RA

Con el objetivo de mejorar las predicciones del modelo y seleccionar la opción más confiable para la clasificación de actividades, se ha utilizado el algoritmo *Fine Tree* como se observa en la FIGURA 82, para entrenar el modelo de inteligencia artificial. Como resultado, se ha observado un incremento en la precisión de las predicciones. En el resumen obtenido, se presentan las métricas del modelo para evaluar y comparar los modelos generados. Las métricas de resultados de entrenamiento se calculan en base al conjunto de validación y muestran una

nueva tasa de efectividad del 96.0%, además de la velocidad de predicción, el tiempo empleado en el entrenamiento y el tamaño del modelo para su exportación. Esto demuestra que la efectividad de la predicción es alta, superando así a los modelos previamente entrenados.

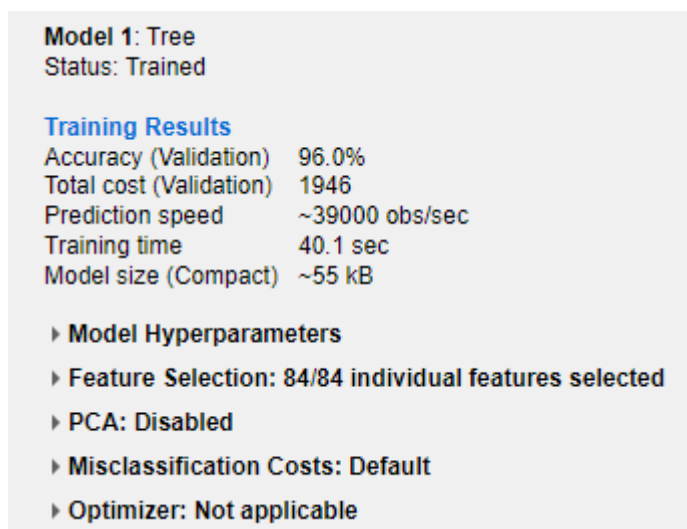


FIGURA 82 Summary Fine Tree RA

En referencia al modelo entrenado de inteligencia artificial llamado *Fine Tree* se observa que en la FIGURA 83 se han analizado los resultados al seleccionar características específicas para mostrar en los ejes X e Y de los predictores. Esto permite una visualización más clara de los resultados para cada clase, brindándonos la opción de mostrar u ocultar clases particulares mediante casillas de verificación para datos correctos e incorrectos. Al examinar la figura correspondiente, se puede apreciar un alto nivel de efectividad en términos de porcentaje.

La visualización gráfica de la matriz de confusión proporciona información sobre el desempeño del clasificador en el modelo *Fine Tree* como lo muestra la FIGURA 84. Es posible observar que la actividad "Stand Up" presentan una precisión de clasificación del 100%. Esto significa que el 100% de los puntos en estas clases están correctamente clasificados, como se refleja en la celda azul de la columna TPR. El porcentaje de clasificación correcta en las actividades a clasificar supera el 90%. Además, la tasa de falsos negativos, es decir, los datos clasificados incorrectamente en estas clases, es mínima, tal como se muestra en la celda naranja de la

columna FNR. En resumen, el algoritmo *Fine Tree* demuestra una alta eficacia para clasificar las actividades de recolección de café en el módulo *Right Arm*.

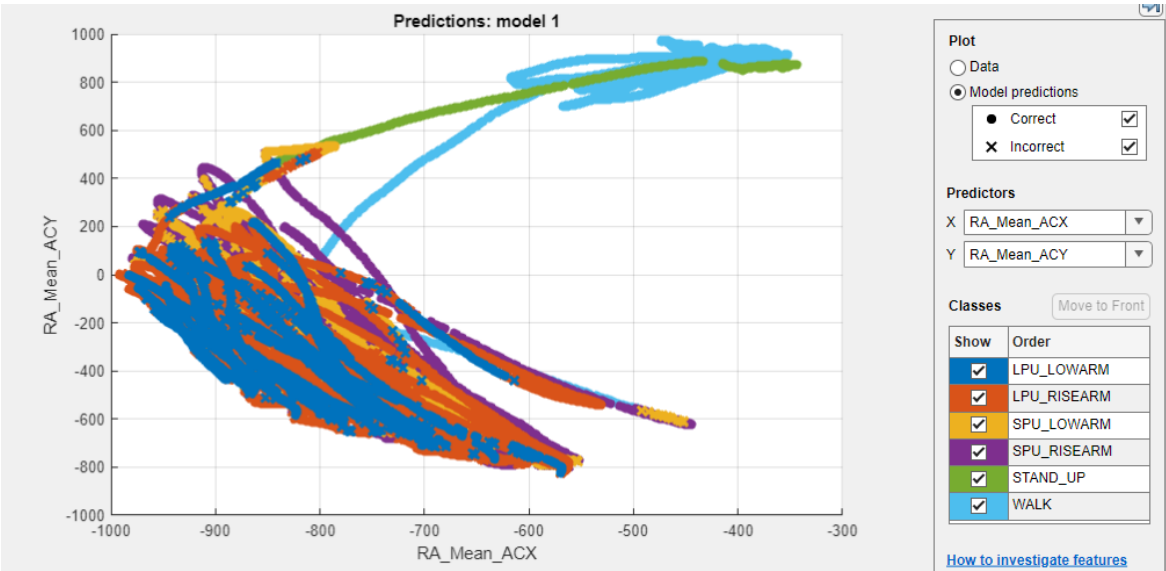


FIGURA 83 Prediction Model Fine Tree RA

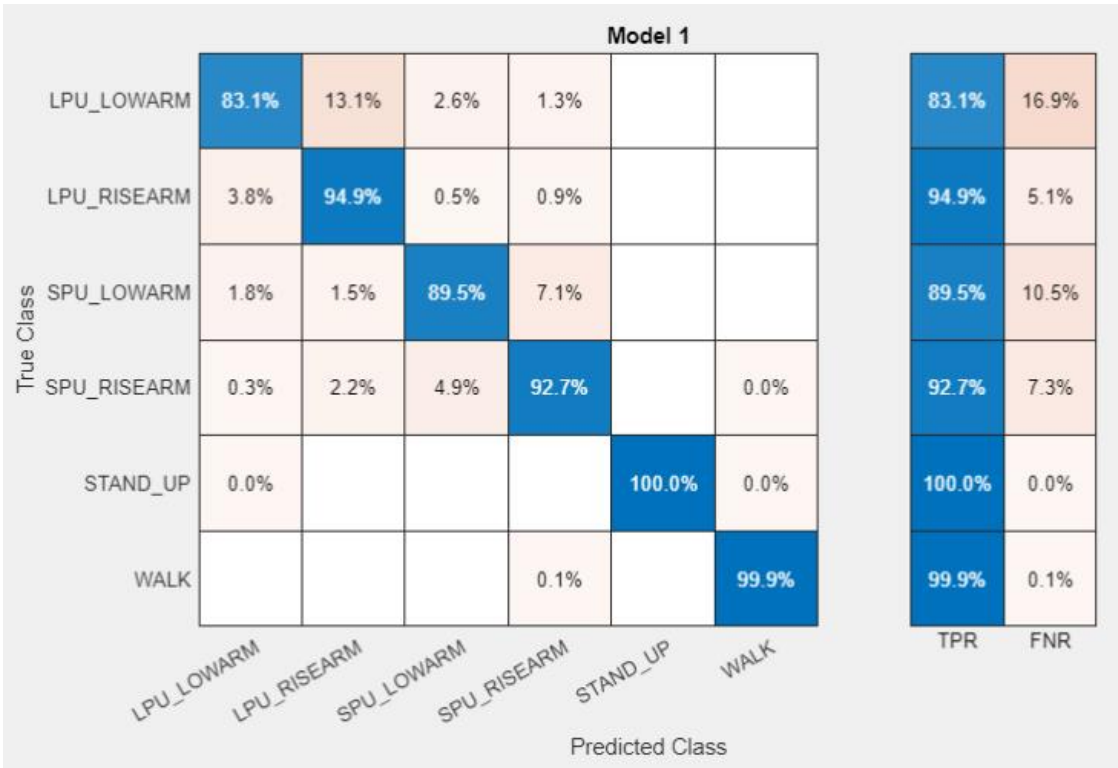


FIGURA 84 Matrix Confusion Fine Tree

La curva de característica de operación (ROC) en la FIGURA 85 proporciona información sobre el rendimiento del algoritmo *Fine Tree*, clasificándolo con alto nivel debido a que los valores son cercanos a 1 y esto indican una clasificación positiva, mientras que los valores alejados de 0 indican una clasificación negativa. Sin embargo, el porcentaje mostrado en la curva no es una medida representativa de la efectividad del algoritmo en sí. Esta efectividad se determina al evaluar la relación entre los porcentajes de valores positivos y negativos, lo que permite predecir si la curva se inclina hacia el punto exacto de clasificación. La curva de ROC del *Fine Tree* se muestra en la gráfica adjunta.

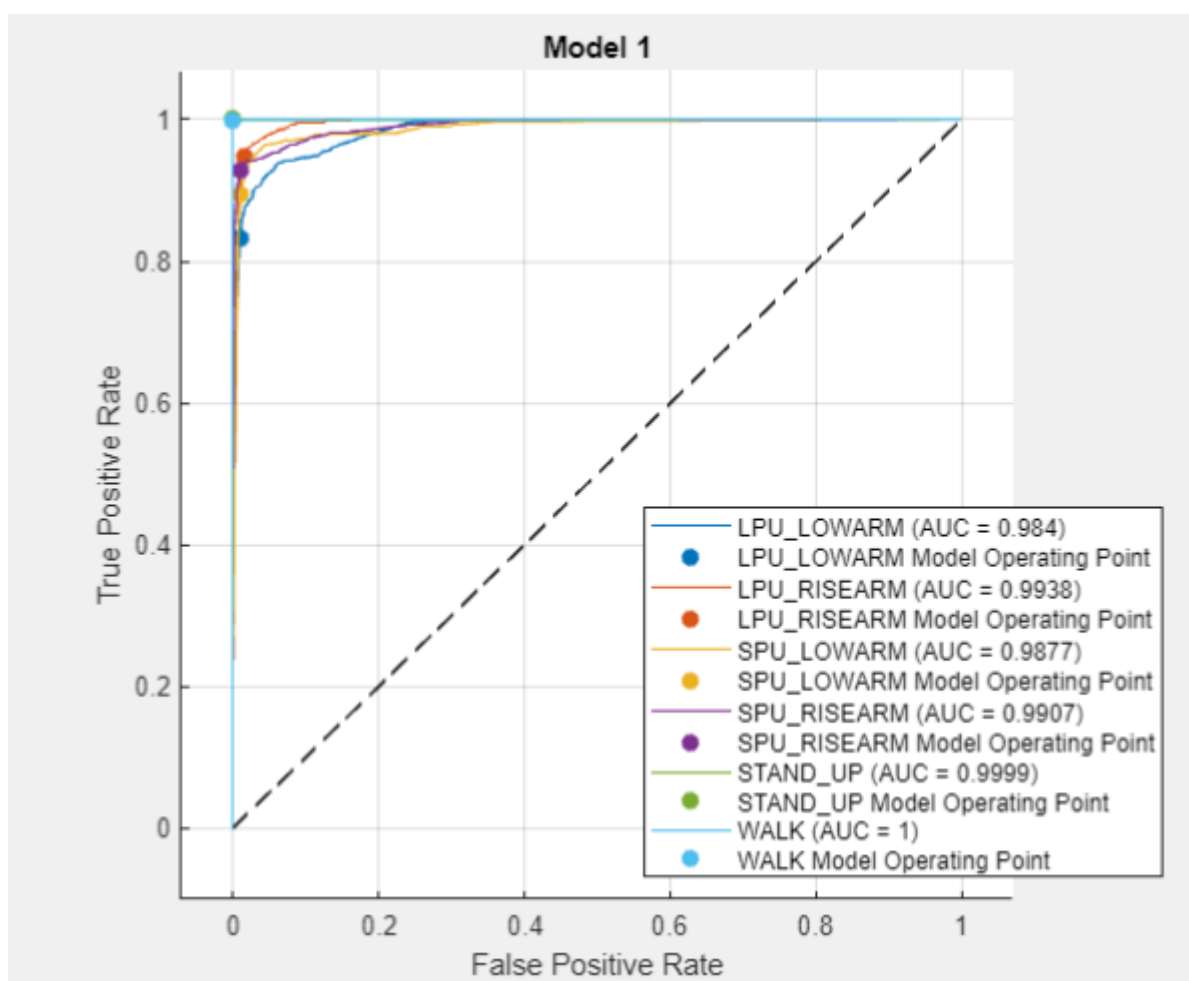


FIGURA 85 ROC Fine Tree RA

5. CONCLUSIONES

El módulo de aplicación *classification learner* permite realizar diferentes modelos de inteligencia artificial que muestran el porcentaje de efectividad y entrenamiento respecto a cada una de las actividades realizadas durante la recolección de café.

Los árboles de decisión presentan una efectividad con mayor precisión en la clasificación de características con los datos entrenados.

Los modelos entrenados de los módulos *Chest*, *Left Arm* *Left Right* que se presentan, tienen una aproximación de eficacia superior al 95% en los diagramas de árbol de decisiones y modelo de IA para el entrenamiento del conjunto de datos, este valor puede variar dependiendo el número de características que extraigan; en base a los resultados obtenidos se puede evidenciar que estos algoritmos *Machine Learning* son los más apropiados para el entrenamiento de algoritmos de clasificación de actividades. El proceso de entrenamiento de algoritmos utiliza *Machine Learning* para agrupar y analizar con mayor precisión una gran cantidad de características extraídas de los datos adquiridos. Esto permite clasificar y analizar los datos de manera más eficiente y efectiva.

El árbol de decisión *Fine Tree* para el módulo *Chest* se muestra en la FIGURA 87 y obtiene una precisión del 97.1%, por lo que el modelo cuenta con características de mayor poder predictivo para clasificar las actividades de manera confiable, como se muestra en la figura la matriz de confusión muestra una diagonal con celdas en color azul indicando los valores verdaderos, y los resultados obtenidos en la columna TPR. Así como también en la curva ROC indica el porcentaje de rendimiento de las actividades clasificadas, esto permiten observar que los puntos del diagrama se inclina el valor de (1) para el caso de las actividades realizadas durante el proceso de recolección de café clasificadas correctamente.

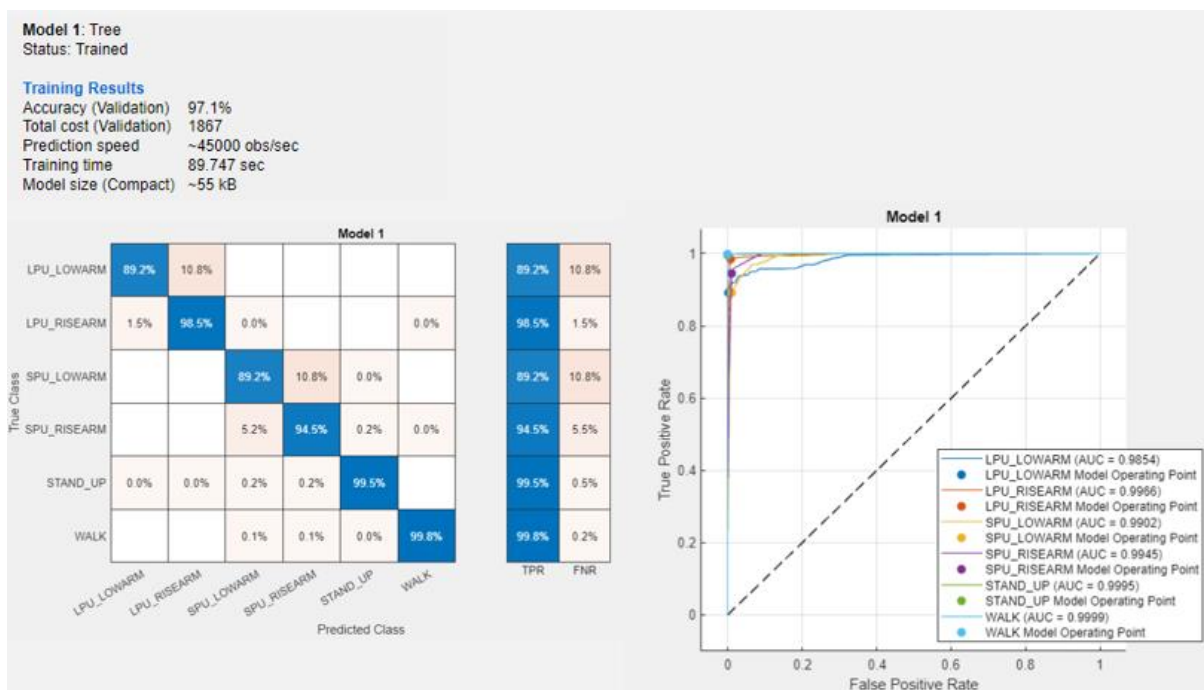


FIGURA 87 Resultados Módulo Chest

Para el módulo *Left Arm* en la FIGURA 88, se observa que el modelo de inteligencia artificial Fine Tree, logra obtener una precisión del 97.1%, por lo que es el más adecuado utilizar. Esto significa que el modelo tiene características que lo hacen altamente efectivo para clasificar las actividades de manera confiable. En la figura adjunta, la matriz de confusión muestra una diagonal con celdas en color azul, lo cual indica los valores verdaderos, y los resultados obtenidos en la columna TPR. Además, la curva ROC muestra el rendimiento porcentual de las actividades clasificadas. Esto nos permite observar que los puntos en la curva ROC se inclinan hacia el valor de (1), lo que indica que las actividades realizadas durante el proceso de recolección de café se clasifican correctamente.

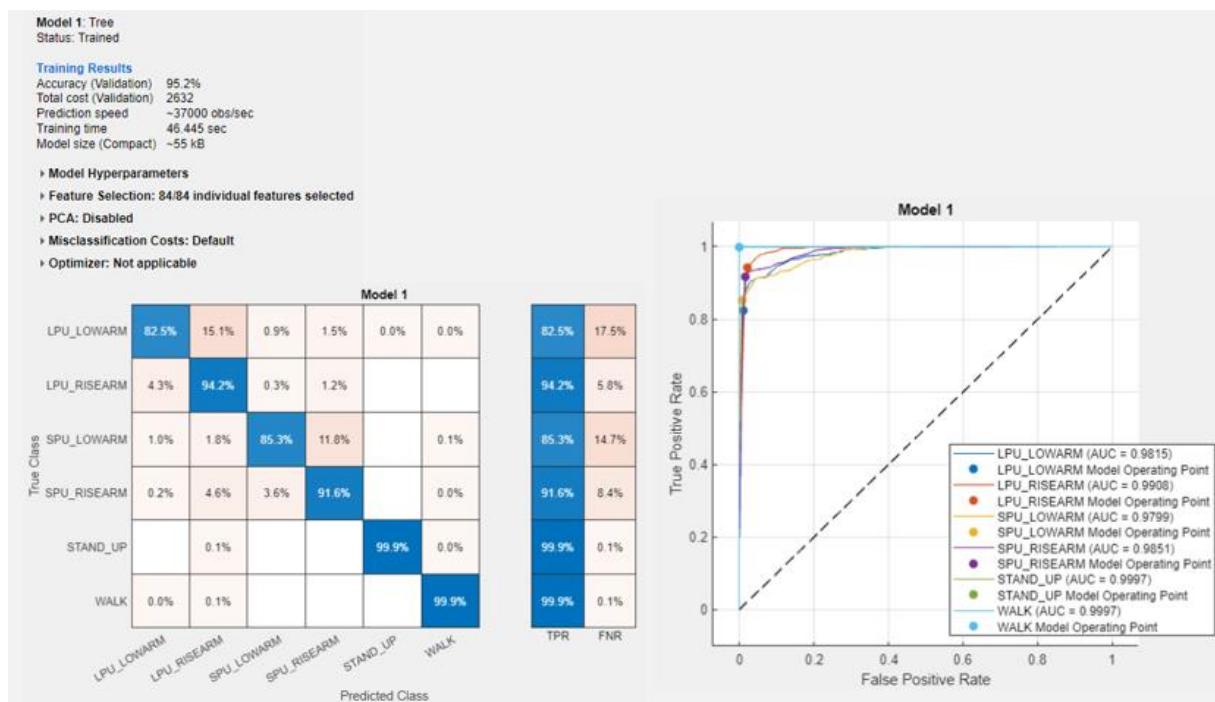


FIGURA 88 Resultados Módulo Left Arm

Para el caso del módulo Right Arm

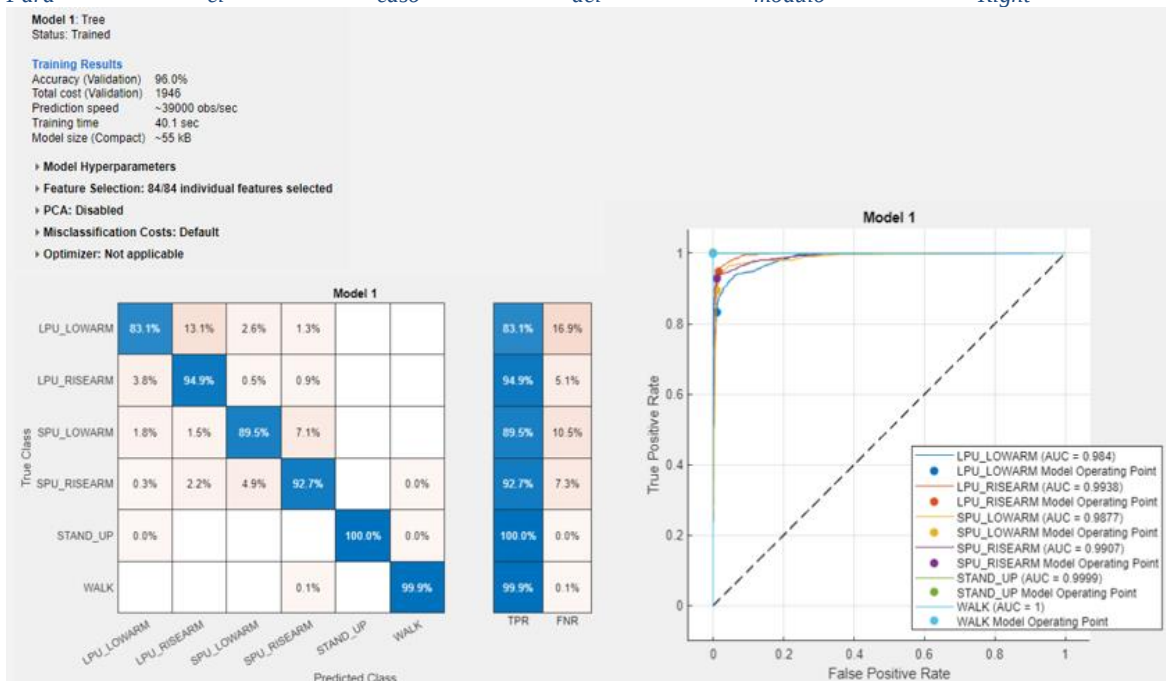
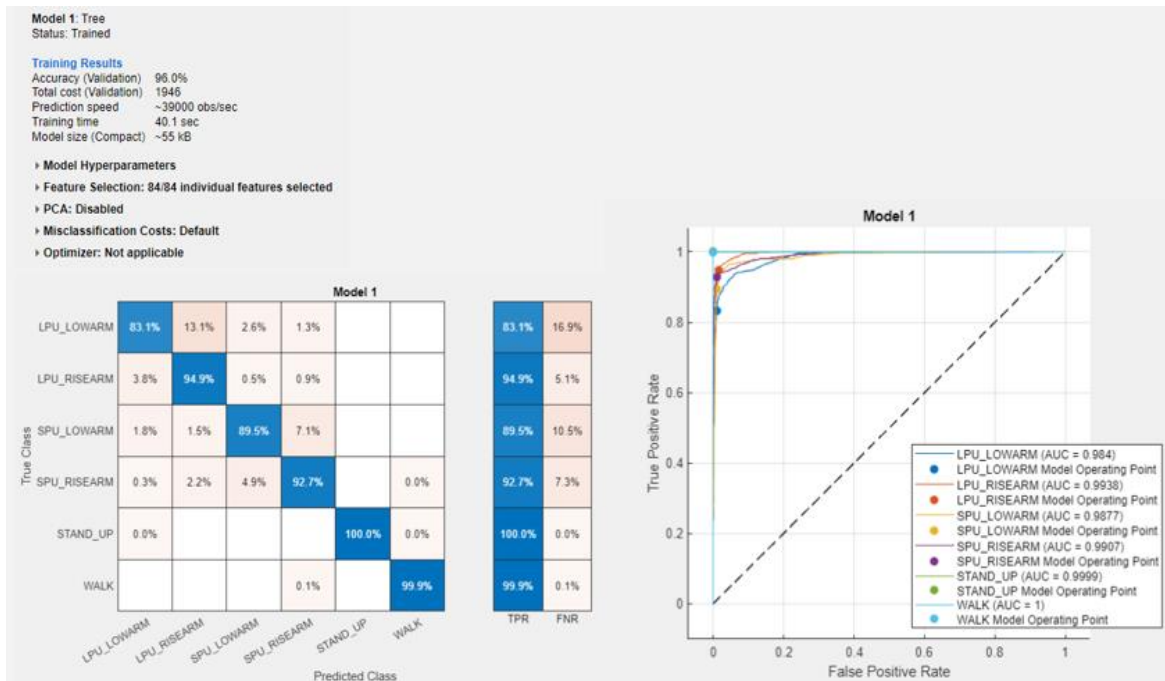


FIGURA 89, se considera el árbol de decisión *Fine Tree* implementado el cual ha demostrado ser altamente preciso, con una tasa de acierto del 96%. Este modelo posee características destacadas que le otorgan un mayor poder predictivo al momento de clasificar las actividades

relacionadas de forma confiable. Al observar la matriz de confusión en la figura, se aprecia la diagonal con celdas en color azul, indicando los valores verdaderos. Los resultados obtenidos en la columna TPR refuerzan aún más la eficacia del modelo. Además, al analizar la curva ROC, se revela el porcentaje de rendimiento de las actividades clasificadas. En este caso, se puede apreciar que los puntos del diagrama tienden hacia el valor de (1), lo cual indica una clasificación precisa de las actividades realizadas durante el proceso de recolección de café. Estos resultados respaldan la confiabilidad y efectividad del modelo de árbol de decisión *Fine Tree* en la clasificación de actividades relacionadas con la recolección de café.



Los modelos de *IA* hacen posible el aprendizaje automático llevando a cabo procesos computacionales eficientes para la detección de las actividades humanas, en conclusión, la inteligencia artificial proporciona un enfoque innovador para mejorar los sistemas de procesamiento de señales en diferentes aplicaciones del mundo real. La utilización de herramientas como MATLAB ofrece la posibilidad de explorar y preprocesar datos de manera interactiva, diseñar características relevantes, crear modelos de *IA* y finalmente implementar sistemas basados en *IA*. Estas capacidades brindan nuevas oportunidades para mejorar la eficiencia y la precisión en el procesamiento de señales, lo que puede tener un impacto significativo en diversas áreas y sectores que requieren análisis y procesamiento de señales avanzados.

REFERENCIAS

- Bajarin T. (2016). *The Real Reason Apple Made the Apple Watch*.
- Bensene R. (n.d.). *Hewlett Packard HP-01 Calculator Wristwatch*. Retrieved March 23, 2022, from <https://www.oldcalculatormuseum.com/hp-01.html>
- Carlos Lopez, J., & Guerra, L. (2016). *Dispositivos y tecnologías wearables*.
- Chen, F. (2021). Athlete muscle measurement and exercise data monitoring based on embedded system and wearable devices. *Microprocessors and Microsystems*, 82(December 2020), 103901. <https://doi.org/10.1016/j.micpro.2021.103901>
- College of Optometrists. (n.d.). Retrieved March 22, 2022, from <https://www.college-optometrists.org/the-college/museum/online-exhibitions/virtual-spectacles-gallery/the-invention-of-spectacles.html>
- Dreiseitl, S., & Ohno-Machado, L. (2002). Logistic regression and artificial neural network classification models: A methodology review. *Journal of Biomedical Informatics*, 35(5–6), 352–359. [https://doi.org/10.1016/S1532-0464\(03\)00034-0](https://doi.org/10.1016/S1532-0464(03)00034-0)
- Golden, R. M. (2020). Formal Machine Learning Algorithms. In *Statistical Machine Learning*. <https://doi.org/10.1201/9781351051507-3>
- Kim J.-H. (2007). *A Study on the Characteristics of Modern Fashion Design for Digital Nomadic Culture*. 6–14.
- Kresser C. (n.d.). *The Benefits of Wearable Technology for Health Tracking | Chris Kresser*. Retrieved March 21, 2022, from <https://chriskresser.com/the-benefits-of-using-wearable-technology-for-health-tracking/>
- Lai, X., Liu, Q., Wei, X., Wang, W., Zhou, G., & Han, G. (2013). A survey of body sensor networks. In *Sensors (Switzerland)* (Vol. 13, Issue 5). <https://doi.org/10.3390/s130505406>

- Luczak, T., Burch, R., Lewis, E., Chander, H., & Ball, J. (n.d.). *State-of-the-art review of athletic wearable technology: What 113 strength and conditioning coaches and athletic trainers from the USA said about technology in sports*. <https://doi.org/10.1177/1747954119885244>
- Marshall G. (2016). *The Story of Fitbit: How a Wooden Box Became a \$4 Billion Company*.
- McClure, K., Erdreich, B., Bates, J. H. T., McGinnis, R. S., Masquelin, A., & Wshah, S. (2020). Classification and detection of breathing patterns with wearable sensors and deep learning. *Sensors (Switzerland)*, 20(22), 1–13. <https://doi.org/10.3390/s20226481>
- Nguyen M. (2016). *The Most Successful Wearables for Consumers*.
- Plantronics J.-H. (2010). *An Industry Pioneer Reflects on a Decade'S Worth of Achievements To Honour the 10th Anniversary of Bluetooth Headsets*.
- Polygon O. (2019). *Watch Timeline | Time killers: The strange history of wrist gaming*. <https://www.polygon.com/a/smartwatch-history-guide-evolution/watch-timeline>
- Python Machine Learning - Vahid Mirjalili, Sebastian Raschka - Google Libros*. (n.d.). Retrieved June 23, 2022, from <https://books.google.es/books?hl=es&lr=&id=5EtOEAAAQBAJ&oi=fnd&pg=PT5&dq=machine+learning+modelo+de+aprendizaje+autom%C3%A1tico&ots=eqJ2QtYHPa&sig=lq0gXF8orMKImoIROnNRpJR7CE#v=onepage&q=machine%20learning%20modelo%20de%20aprendizaje%20autom%C3%A1tico&f=false>
- Sangwan, A., & Bhattacharya, P. P. (2015). Wireless Body Sensor Networks: A Review. *International Journal of Hybrid Information Technology*, 8(9), 105–120. <https://doi.org/10.14257/ijhit.2015.8.9.12>
- Sattar, F., Azad, A. A. M., Sikder, M. S., & Arefin, M. S. (2019). Body sensor networks for monitoring performances in sports: A brief overview and some new thoughts. *Artificial Intelligence Research*, 8(1), 25. <https://doi.org/10.5430/air.v8n1p25>
- Shalev-Shwartz, S., & Ben-David, S. (2013). Understanding machine learning: From theory to algorithms. In *Understanding Machine Learning: From Theory to Algorithms* (Vol. 9781107057). <https://doi.org/10.1017/CBO9781107298019>

- Situnayake, D. (2020). *TinyML*. <https://www.oreilly.com/library/view/tinyml/9781492052036/>
- Starr M. (2014). *Tommy Hilfiger Launches Solar Power Jackets To Charge Your Phone*.
- The 2017 WT Wearable Technologies*. (n.d.). Retrieved October 20, 2022, from <https://www.innovationworldcup.com/8192/>
- The National Archives. (n.d.). *Fighting talk: First World War telecommunications - The National Archives*. Retrieved March 23, 2022, from <https://www.nationalarchives.gov.uk/first-world-war/telecommunications-in-war/>
- Trastornos musculoesqueléticos*. (n.d.). Retrieved January 5, 2022, from <https://www.who.int/es/news-room/fact-sheets/detail/musculoskeletal-conditions>
- Aroganam G, Manivannan N, Harrison D. Review on Wearable Technology Sensors Used in Consumer Sport Applications. *Sensors* (Basel). 2019 Apr 28;19(9) 1983.<https://doi.org/10.3390/s19091983>

APÉNDICE

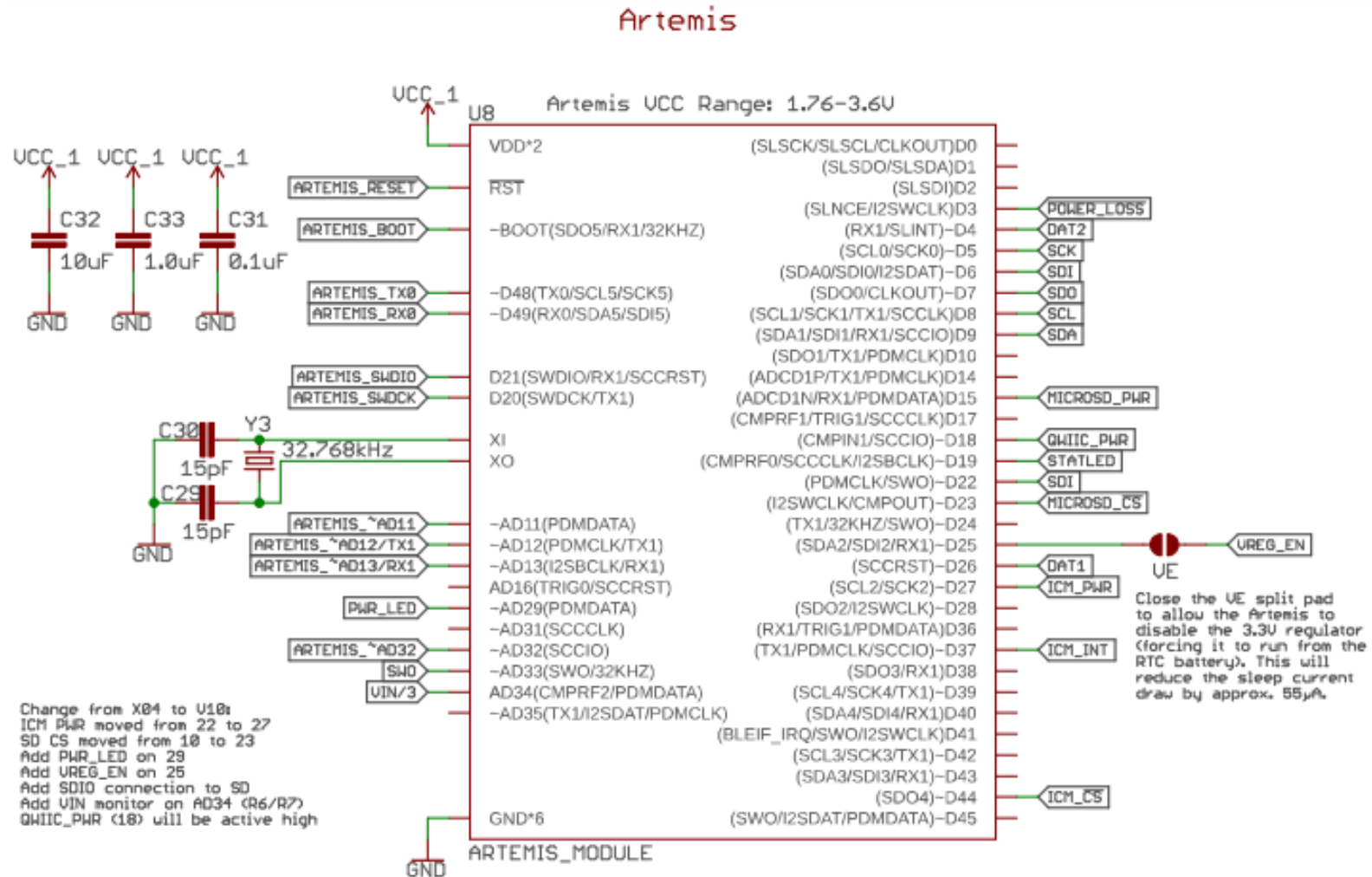
Características del Giroscopio

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
GYROSCOPE SENSITIVITY						
Full-Scale Range	GYRO_FS_SEL=0		±250		dps	1
	GYRO_FS_SEL=1		±500		dps	1
	GYRO_FS_SEL=2		±1000		dps	1
	GYRO_FS_SEL=3		±2000		dps	1
Gyroscope ADC Word Length			16		bits	1
Sensitivity Scale Factor	GYRO_FS_SEL=0		131		LSB/(dps)	1
	GYRO_FS_SEL=1		65.5		LSB/(dps)	1
	GYRO_FS_SEL=2		32.8		LSB/(dps)	1
	GYRO_FS_SEL=3		16.4		LSB/(dps)	1
Sensitivity Scale Factor Tolerance	25°C		±1.5		%	2
Sensitivity Scale Factor Variation Over Temperature	-40°C to +85°C		±3		%	2
Nonlinearity	Best fit straight line; 25°C		±0.1		%	2, 3
Cross-Axis Sensitivity			±2		%	2, 3
ZERO-RATE OUTPUT (ZRO)						
Initial ZRO Tolerance	25°C (Component-level)		±5		dps	2
ZRO Variation Over Temperature	-40°C to +85°C		±0.05		dps/°C	2
GYROSCOPE NOISE PERFORMANCE (GYRO_FS_SEL=0)						
Noise Spectral Density	Based on Noise Bandwidth = 10 Hz		0.015		dps/√Hz	2
GYROSCOPE MECHANICAL FREQUENCIES		25	27	29	kHz	2
LOW PASS FILTER RESPONSE	Programmable Range	5.7		197	Hz	1, 3
GYROSCOPE START-UP TIME	From Full-Chip Sleep mode		35		ms	2, 3
OUTPUT DATA RATE	Low-Power Mode	4.4		562.5	Hz	1
	Low-Noise Mode GYRO_FCHOICE=1; GYRO_DLPFCFG=x	4.4		1.125k	Hz	
	Low-Noise Mode GYRO_FCHOICE=0; GYRO_DLPFCFG=x			9k	Hz	

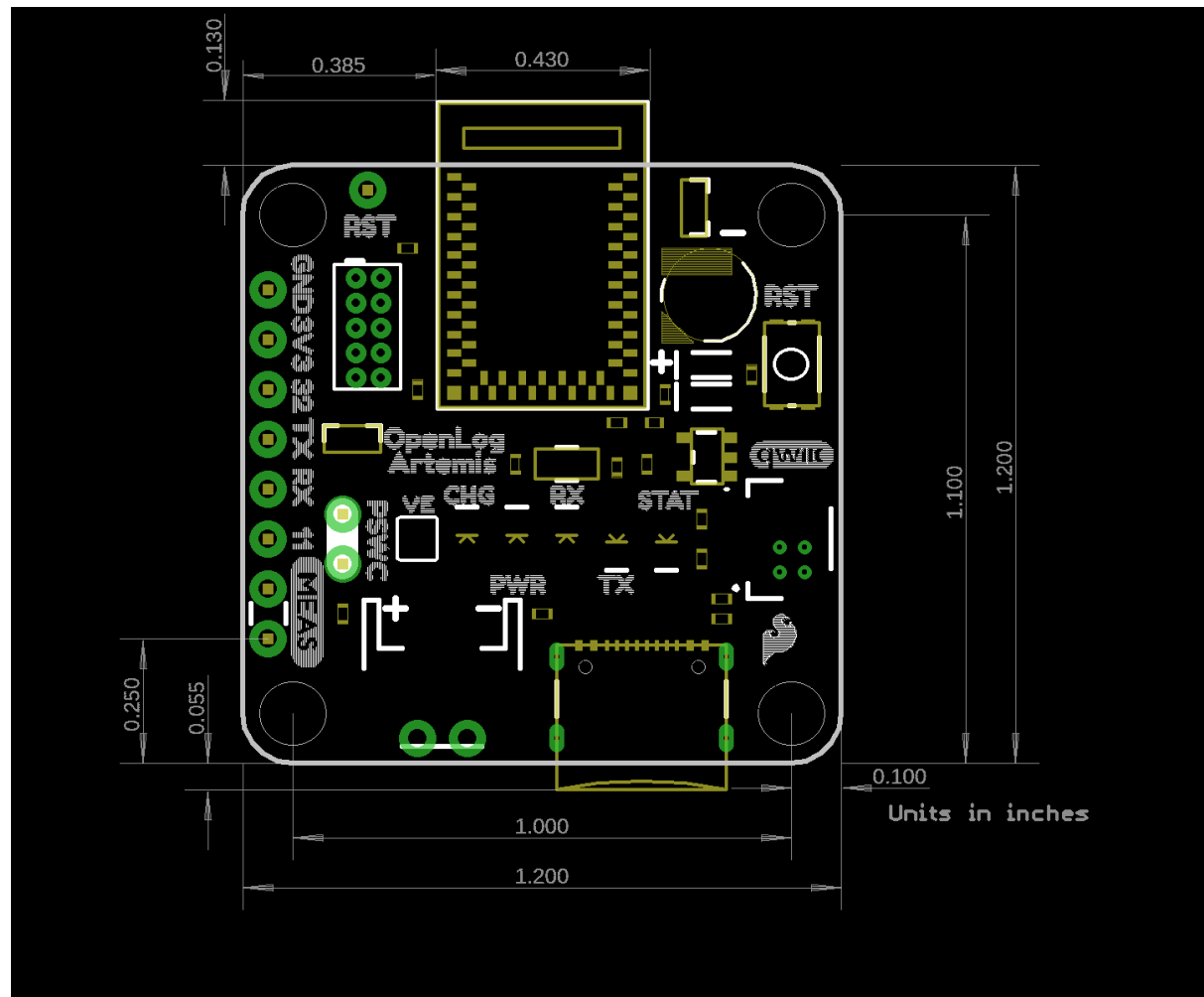
Características del Acelerómetro

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
ACCELEROMETER SENSITIVITY						
Full-Scale Range	ACCEL_FS=0		±2		G	1
	ACCEL_FS=1		±4		G	1
	ACCEL_FS=2		±8		G	1
	ACCEL_FS=3		±16		G	1
ADC Word Length	Output in two's complement format		16		Bits	1
Sensitivity Scale Factor	ACCEL_FS=0		16,384		LSB/g	1
	ACCEL_FS=1		8,192		LSB/g	1
	ACCEL_FS=2		4,096		LSB/g	1
	ACCEL_FS=3		2,048		LSB/g	1
Initial Tolerance	Component-level		±0.5		%	2
Sensitivity Change vs. Temperature	-40°C to +85°C ACCEL_FS=0		±0.026		%/°C	2
Nonlinearity	Best Fit Straight Line		±0.5		%	2, 3
Cross-Axis Sensitivity			±2		%	2, 3
ZERO-G OUTPUT						
Initial Tolerance	Component-level, all axes		±25		mg	2
Initial Tolerance	Board-level, all axes		±50		mg	2
Zero-G Level Change vs. Temperature	0°C to +85°C		±0.80		mg/°C	2
ACCELEROMETER NOISE PERFORMANCE						
Noise Spectral Density	Based on Noise Bandwidth = 10 Hz		230		μg/√Hz	2
LOW PASS FILTER RESPONSE	Programmable Range	5.7		246	Hz	1, 3
ACCELEROMETER STARTUP TIME	From Sleep mode		20		ms	2, 3
	From Cold Start, 1 ms V _{DD} ramp		30		ms	2, 3
OUTPUT DATA RATE	Low-Power Mode	0.27		562.5	Hz	1
	Low-Noise Mode ACCEL_FCHOICE=1; ACCEL_DLPFCFG=x	4.5		1.125k	Hz	
	Low-Noise Mode ACCEL_FCHOICE=0; ACCEL_DLPFCFG=x			4.5k	Hz	

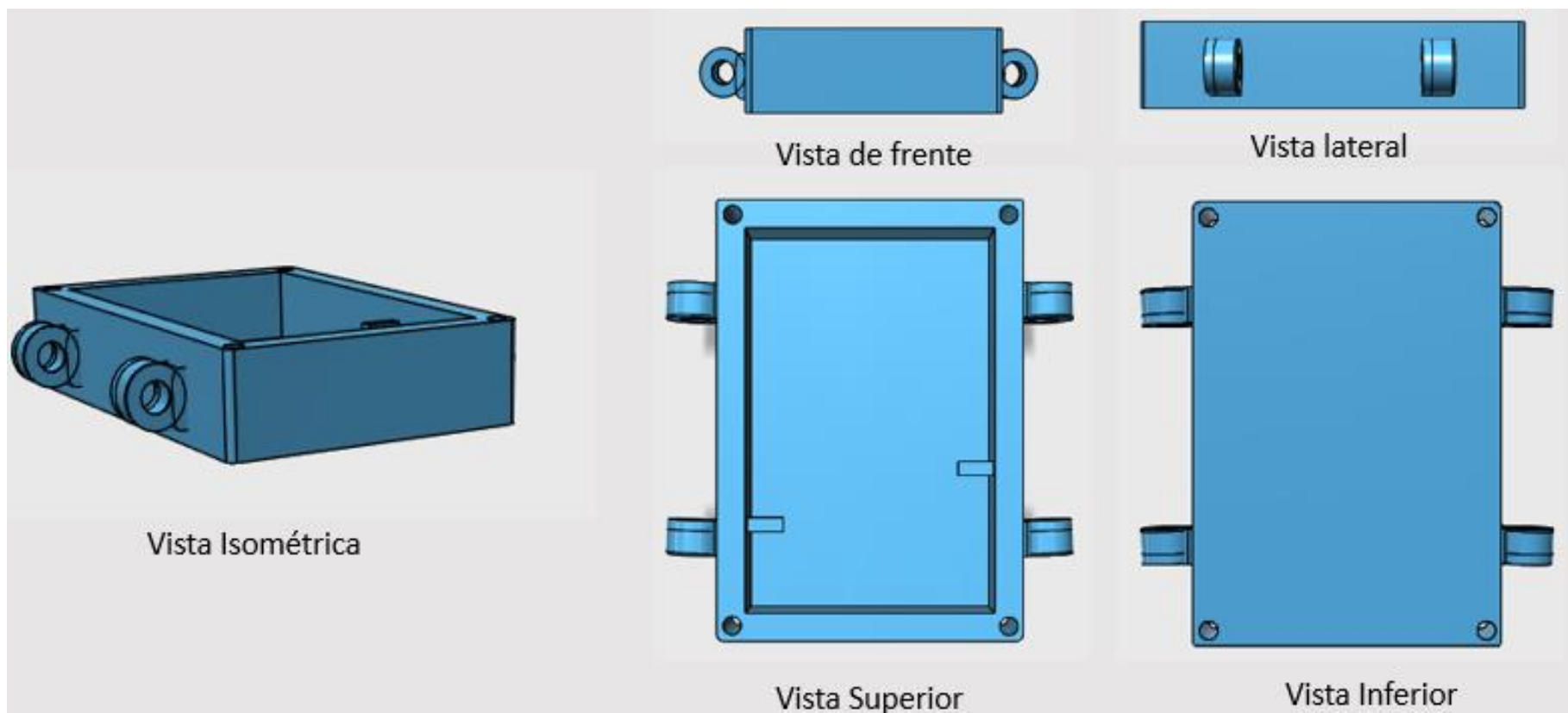
Esquemático sensor OpenLog Artemis



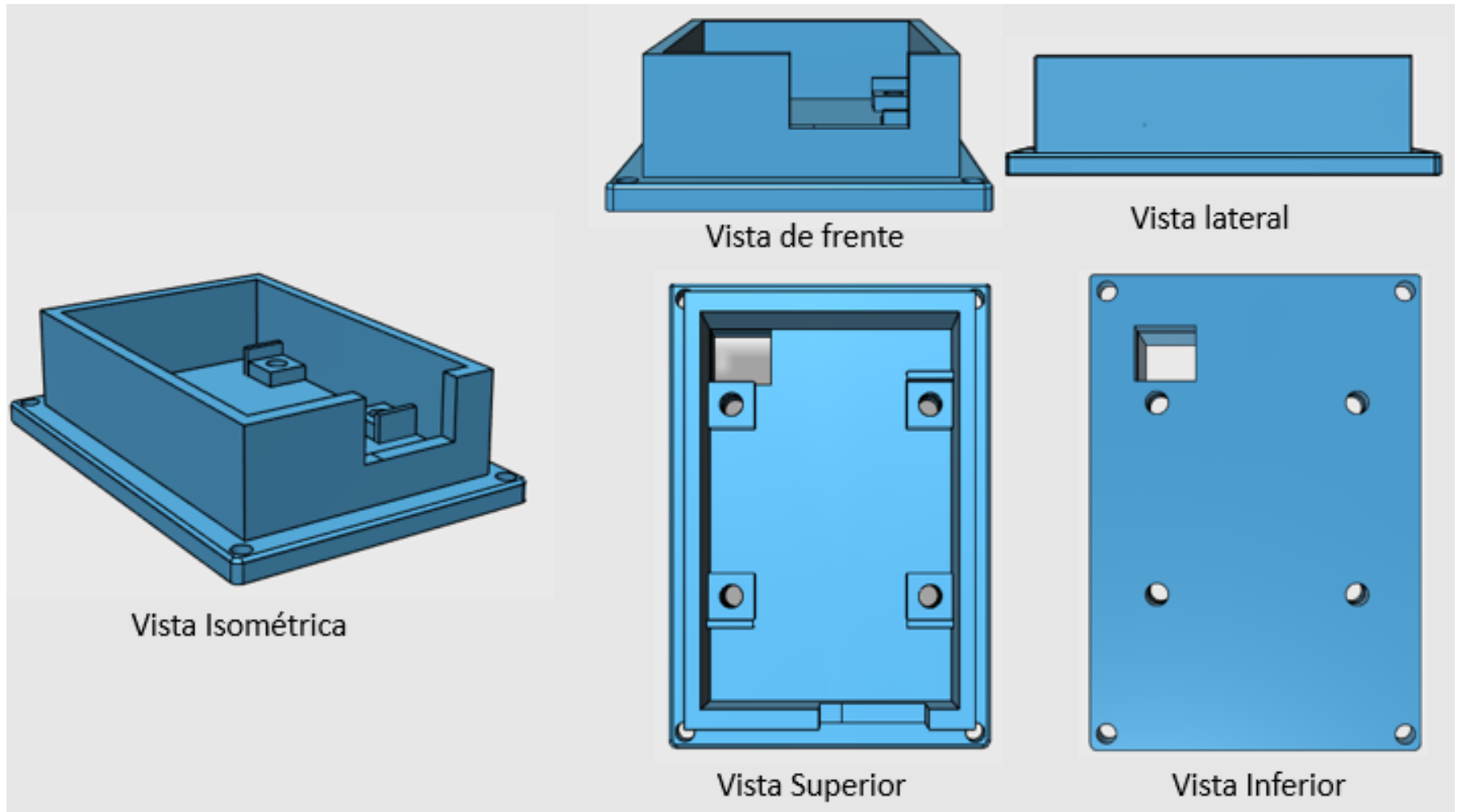
Dimensiones de Sensor OpenLog Artemis



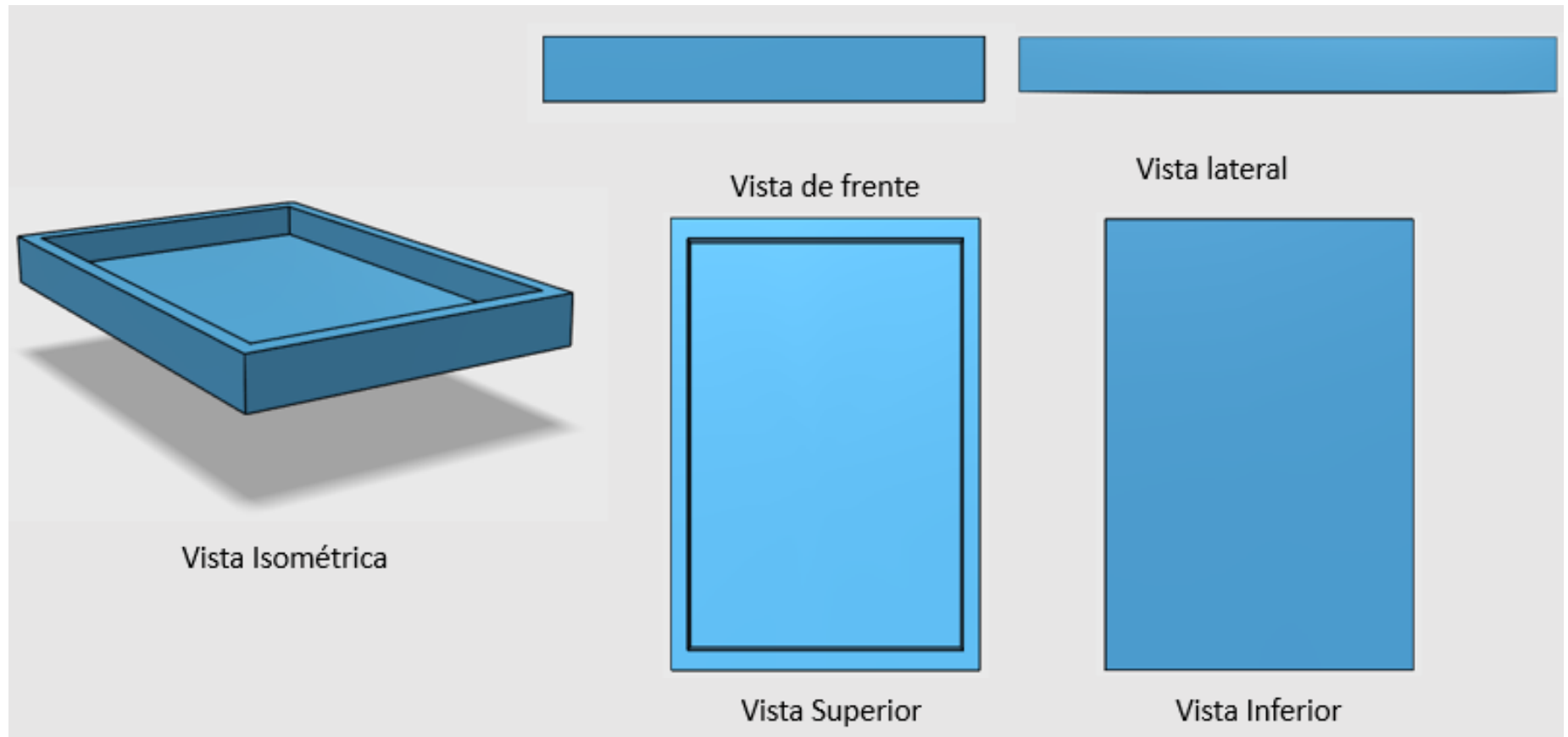
Base para batería del Wearable Device Modelo 3D



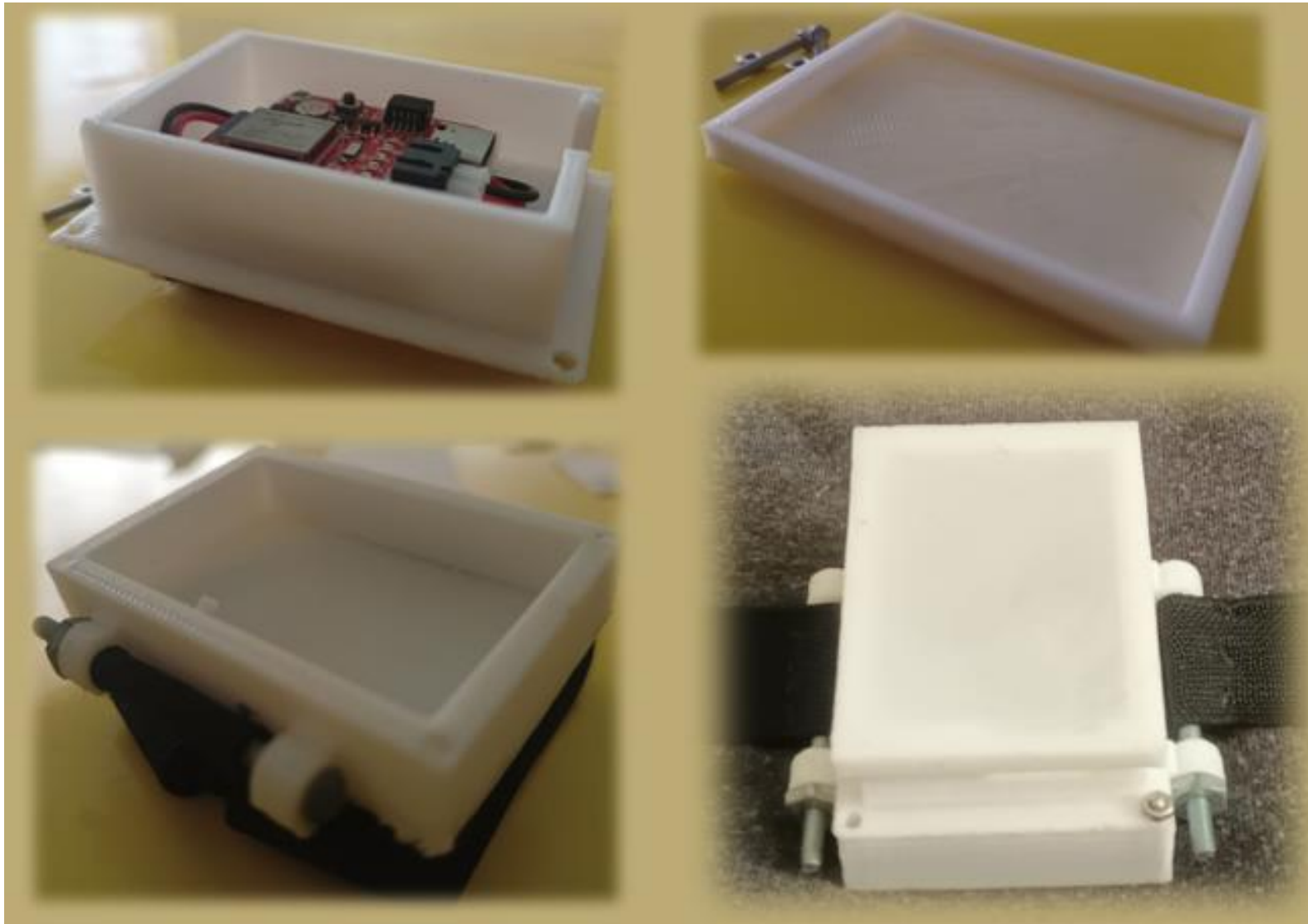
Base para sensor del Wearable Device Modelo 3D



Cubierta superior del Wearable Device Impresión 3D



Módulo Chest Impresión 3D



Módulo Left Arm Impresión 3D



Módulo Right Arm Impresión 3D



ANEXOS

Código para Aplicación en Android

Region Module Attributes

```
#FullScreen: True
#IncludeTitle: False
#ApplicationLabel: PHD:MOV-TTS-TXT
#VersionCode: 1
#VersionName:
#SupportedOrientations: portrait
#CanInstallToExternalStorage: False
#End Region
```

Sub Process_Globals

```
'TEMPORIZADORES
Dim reloj As Timer
'Actualiza la fecha y hora
Dim timer1 As Timer
'Almacena la accion cada segundo
Dim TTS1 As TTS
End Sub
```

Sub Globals

```
'Vibrador
'Dim phv As PhoneVibrate

'VARIABLES: etiqueta de tiempo/acción y
control
Dim cambio As Boolean
Dim año, mes, dia As Int
Dim hora, minuto, segundo,
contadorcambio, contadortts As Int
Dim accionactual, accion, fecha As String
contadorcambio=0
contadortts=0

'VARIABLES: Listas
Dim ListAccionesSegundo As List
ListAccionesSegundo.initialize

'Elementos de interfaz
Private BAccion1 As Button
Private BAccion2 As Button
Private BAccion3 As Button
Private BAccion4 As Button
Private BAccion5 As Button
Private BAccion6 As Button
```

```
Private BAccion7 As Button
Private BAccion8 As Button

Private BAccion9 As Button
Private BAccion10 As Button
Private BAccion11 As Button
Private BAccion12 As Button
Private BCambio As Button
Private BAyuda As Button
Private Encender As Button
Private ETTiempo As EditText
Private ETRenombrar As EditText
Private LFecha As Label
Private LHora As Label
Private LAccionActual As Label
```

End Sub

Sub Activity_Create(FirstTime As Boolean)

```
If FirstTime Then
    Activity.LoadLayout("main")
    Encender.Color=Colors.Gray
    BAccion1.Color=Colors.Gray
    BAccion2.Color=Colors.Gray
    BAccion3.Color=Colors.Gray
    BAccion4.Color=Colors.Gray
    BAccion5.Color=Colors.Gray
    BAccion6.Color=Colors.Gray
    BAccion7.Color=Colors.Gray
    BAccion8.Color=Colors.Gray
    BAccion9.Color=Colors.Gray
    BAccion10.Color=Colors.Gray
    BAccion11.Color=Colors.Gray
    BAccion12.Color=Colors.Gray
End If
reloj.Initialize("reloj",1)
reloj.Enabled=True
```

End Sub

Sub Activity_Resume

```
timer1.Enabled=False
reloj.Enabled=True
contadorcambio=0
contadortts=0
Encender.Color=Colors.Gray
```

```

BAccion1.Color=Colors.Gray
BAccion2.Color=Colors.Gray
BAccion3.Color=Colors.Gray
BAccion4.Color=Colors.Gray
BAccion5.Color=Colors.Gray
BAccion6.Color=Colors.Gray
BAccion7.Color=Colors.Gray
BAccion8.Color=Colors.Gray
BAccion9.Color=Colors.Gray
BAccion10.Color=Colors.Gray
BAccion11.Color=Colors.Gray
BAccion12.Color=Colors.Gray

If TTS1.IsInitialized = False Then
    TTS1.Initialize("TTS1")
End If
End Sub

Sub Activity_Pause (UserClosed As Boolean)
    timer1.Enabled=False
    reloj.Enabled=False
    Encender.Color=Colors.Gray
    BAccion1.Color=Colors.Gray
    BAccion2.Color=Colors.Gray
    BAccion3.Color=Colors.Gray
    BAccion4.Color=Colors.Gray
    BAccion5.Color=Colors.Gray
    BAccion6.Color=Colors.Gray
    BAccion7.Color=Colors.Gray
    BAccion8.Color=Colors.Gray
    BAccion9.Color=Colors.Gray
    BAccion10.Color=Colors.Gray
    BAccion11.Color=Colors.Gray
    BAccion12.Color=Colors.Gray

    contadorcambio=0
    contadortts=0

    If(UserClosed) Then
        TTS1.Release
        timer1.Enabled = False
    End If
End Sub

'Inicia la adquisicion
Sub Encender_Click
    timer1.Initialize("timer1",1000)
    timer1.Enabled=True

    contadorcambio=0

    contadortts=0
    Encender.Color=Colors.Green
    ToastMessageShow("Registro y TTS
    activos",False)
    End Sub

'Finaliza la adquisicion
Sub Encender_LongClick
    fecha= DateTime.GetYear(DateTime.Now)
    & "-" & DateTime.GetMonth(DateTime.Now) & "-"
    & DateTime.GetDayOfMonth(DateTime.Now) & " "
    & DateTime.GetHour(DateTime.Now) & "." &
    DateTime.GetMinute(DateTime.Now) & "." &
    DateTime.GetSecond(DateTime.Now)
    timer1.Enabled=False
    File.Writestring(File.DirRootExternal, fecha
    & "-Etiquetas.txt", ListAccionesSegundo)
    ToastMessageShow("Registro TXT
    DESactivado",False)
    contadorcambio=0
    contadortts=0

    'Reinicia los colores de los botones
    Encender.Color=Colors.Gray
    BAccion1.Color=Colors.Gray
    BAccion2.Color=Colors.Gray
    BAccion3.Color=Colors.Gray
    BAccion4.Color=Colors.Gray
    BAccion5.Color=Colors.Gray
    BAccion6.Color=Colors.Gray
    BAccion7.Color=Colors.Gray
    BAccion8.Color=Colors.Gray
    BAccion9.Color=Colors.Gray
    BAccion10.Color=Colors.Gray
    BAccion11.Color=Colors.Gray
    BAccion12.Color=Colors.Gray
End Sub

'Actualiza la fecha y hora del reloj
Sub reloj_tick
    reloj.Enabled=False
    LFecha.Text=DateTime.GetYear(DateTime.
    Now) & "-" & DateTime.GetMonth(DateTime.Now)
    & "-" & DateTime.GetDayOfMonth(DateTime.Now)
    LHora.text=
    DateTime.GetHour(DateTime.Now) & ":" &
    DateTime.GetMinute(DateTime.Now) & ":" &
    DateTime.GetSecond(DateTime.Now)
    reloj.Enabled=True
End Sub

'Almacena cada segundo la accion

```

```

Sub timer1_tick
    'Almacena cada segundo la accion
    realizada
    timer1.Enabled=False

    'Controla TTS y cambio de accion

    If contadorcambio>=0*60*ETTiempo.Text
Then
        accion=BAccion1.Text
        LAccionActual.Text=accion
        BAccion1.Color=Colors.Green
        BAccion2.Color=Colors.Gray
        BAccion3.Color=Colors.Gray
        BAccion4.Color=Colors.Gray
        BAccion5.Color=Colors.Gray
        BAccion6.Color=Colors.Gray
        BAccion7.Color=Colors.Gray
        BAccion8.Color=Colors.Gray
        BAccion9.Color=Colors.Gray
        BAccion10.Color=Colors.Gray
        BAccion11.Color=Colors.Gray
        BAccion12.Color=Colors.Gray
    End If

    If contadorcambio>=1*60*ETTiempo.Text
Then
        accion=BAccion2.Text
        LAccionActual.Text=accion
        BAccion1.Color=Colors.Gray
        BAccion2.Color=Colors.Green
        BAccion3.Color=Colors.Gray
        BAccion4.Color=Colors.Gray
        BAccion5.Color=Colors.Gray
        BAccion6.Color=Colors.Gray
        BAccion7.Color=Colors.Gray
        BAccion8.Color=Colors.Gray
        BAccion9.Color=Colors.Gray
        BAccion10.Color=Colors.Gray
        BAccion11.Color=Colors.Gray
        BAccion12.Color=Colors.Gray
    End If

    If contadorcambio>=2*60*ETTiempo.Text
Then
        accion=BAccion3.Text
        LAccionActual.Text=accion
        BAccion1.Color=Colors.Gray
        BAccion2.Color=Colors.Gray
        BAccion3.Color=Colors.Green
        BAccion4.Color=Colors.Gray
        BAccion5.Color=Colors.Gray

```

```

        BAccion6.Color=Colors.Gray
        BAccion7.Color=Colors.Gray
        BAccion8.Color=Colors.Gray
        BAccion9.Color=Colors.Gray
        BAccion10.Color=Colors.Gray
        BAccion11.Color=Colors.Gray
        BAccion12.Color=Colors.Gray
    End If

    If contadorcambio>=3*60*ETTiempo.Text
Then
        accion=BAccion4.Text
        LAccionActual.Text=accion
        BAccion1.Color=Colors.Gray
        BAccion2.Color=Colors.Gray
        BAccion3.Color=Colors.Gray
        BAccion4.Color=Colors.Green
        BAccion5.Color=Colors.Gray
        BAccion6.Color=Colors.Gray
        BAccion7.Color=Colors.Gray
        BAccion8.Color=Colors.Gray
        BAccion9.Color=Colors.Gray
        BAccion10.Color=Colors.Gray
        BAccion11.Color=Colors.Gray
        BAccion12.Color=Colors.Gray
    End If

    If contadorcambio>=4*60*ETTiempo.Text
Then
        accion=BAccion5.Text
        LAccionActual.Text=accion
        BAccion1.Color=Colors.Gray
        BAccion2.Color=Colors.Gray
        BAccion3.Color=Colors.Gray
        BAccion4.Color=Colors.Gray
        BAccion5.Color=Colors.Green
        BAccion6.Color=Colors.Gray
        BAccion7.Color=Colors.Gray
        BAccion8.Color=Colors.Gray
        BAccion9.Color=Colors.Gray
        BAccion10.Color=Colors.Gray
        BAccion11.Color=Colors.Gray
        BAccion12.Color=Colors.Gray
    End If

    If contadorcambio>=5*60*ETTiempo.Text
Then
        accion=BAccion6.Text
        LAccionActual.Text=accion
        BAccion1.Color=Colors.Gray
        BAccion2.Color=Colors.Gray
        BAccion3.Color=Colors.Gray

```

	BAccion4.Color=Colors.Gray BAccion5.Color=Colors.Gray BAccion6.Color=Colors.Green BAccion7.Color=Colors.Gray BAccion8.Color=Colors.Gray BAccion9.Color=Colors.Gray BAccion10.Color=Colors.Gray BAccion11.Color=Colors.Gray BAccion12.Color=Colors.Gray End If		BAccion2.Color=Colors.Gray BAccion3.Color=Colors.Gray BAccion4.Color=Colors.Gray BAccion5.Color=Colors.Gray BAccion6.Color=Colors.Gray BAccion7.Color=Colors.Gray BAccion8.Color=Colors.Gray BAccion9.Color=Colors.Green BAccion10.Color=Colors.Gray BAccion11.Color=Colors.Gray BAccion12.Color=Colors.Gray End If
Then	If contadorcambio>=6*60*ETTiempo.Text accion=BAccion7.Text LAccionActual.Text=accion BAccion1.Color=Colors.Gray BAccion2.Color=Colors.Gray BAccion3.Color=Colors.Gray BAccion4.Color=Colors.Gray BAccion5.Color=Colors.Gray BAccion6.Color=Colors.Gray BAccion7.Color=Colors.Green BAccion8.Color=Colors.Gray BAccion9.Color=Colors.Gray BAccion10.Color=Colors.Gray BAccion11.Color=Colors.Gray BAccion12.Color=Colors.Gray End If	Then	If contadorcambio>=9*60*ETTiempo.Text accion=BAccion10.Text LAccionActual.Text=accion BAccion1.Color=Colors.Gray BAccion2.Color=Colors.Gray BAccion3.Color=Colors.Gray BAccion4.Color=Colors.Gray BAccion5.Color=Colors.Gray BAccion6.Color=Colors.Gray BAccion7.Color=Colors.Gray BAccion8.Color=Colors.Gray BAccion9.Color=Colors.Gray BAccion10.Color=Colors.Green BAccion11.Color=Colors.Gray BAccion12.Color=Colors.Gray End If
Then	If contadorcambio>=7*60*ETTiempo.Text accion=BAccion8.Text LAccionActual.Text=accion BAccion1.Color=Colors.Gray BAccion2.Color=Colors.Gray BAccion3.Color=Colors.Gray BAccion4.Color=Colors.Gray BAccion5.Color=Colors.Gray BAccion6.Color=Colors.Gray BAccion7.Color=Colors.Gray BAccion8.Color=Colors.Green BAccion9.Color=Colors.Gray BAccion10.Color=Colors.Gray BAccion11.Color=Colors.Gray BAccion12.Color=Colors.Gray End If	If	contadorcambio>=10*60*ETTiempo.Text Then accion=BAccion11.Text LAccionActual.Text=accion BAccion1.Color=Colors.Gray BAccion2.Color=Colors.Gray BAccion3.Color=Colors.Gray BAccion4.Color=Colors.Gray BAccion5.Color=Colors.Gray BAccion6.Color=Colors.Gray BAccion7.Color=Colors.Gray BAccion8.Color=Colors.Gray BAccion9.Color=Colors.Gray BAccion10.Color=Colors.Gray BAccion11.Color=Colors.Green BAccion12.Color=Colors.Gray End If
Then	If contadorcambio>=8*60*ETTiempo.Text accion=BAccion9.Text LAccionActual.Text=accion BAccion1.Color=Colors.Gray	End If	If contadorcambio>=11*60*ETTiempo.Text Then accion=BAccion12.Text

```

        LAccionActual.Text=accion
        BAccion1.Color=Colors.Gray
        BAccion2.Color=Colors.Gray
        BAccion3.Color=Colors.Gray
        BAccion4.Color=Colors.Gray
        BAccion5.Color=Colors.Gray
        BAccion6.Color=Colors.Gray
        BAccion7.Color=Colors.Gray
        BAccion8.Color=Colors.Gray
        BAccion9.Color=Colors.Gray
        BAccion10.Color=Colors.Gray
        BAccion11.Color=Colors.Gray
        BAccion12.Color=Colors.Green
    End If
,
    'Checa si se presiono el boton de accion
    sin clasificar
    If(cambio==True) Then
        accion="Unclassified"
        LAccionActual.Text=accion
    End If

    'Modificacion: Dice cada 10 seg la accion a
    realizar
    If contadortts==9 Then
        TTS1.Speak(accion, True)
        contadortts=0
    End If

    'Guardar en TXT las acciones con fecha

    accionactual=
    DateTime.GetYear(DateTime.Now) & "-" &
    DateTime.GetMonth(DateTime.Now) & "-" &
    DateTime.GetDayOfMonth(DateTime.Now) & " " &
    DateTime.GetHour(DateTime.Now) & ":" &
    DateTime.GetMinute(DateTime.Now) & ":" &
    DateTime.GetSecond(DateTime.Now) & TAB &
    accion & Chr(13)
    ListAccionesSegundo.Add(accionactual)

    contadortts=contadortts+1
    contadorcambio=contadorcambio+1
    timer1.Enabled = True

    'Se coloca aqui la condicion de paro para
    que si se cumple se detenga el timer
    If
    contadorcambio>=12*60*ETTiempo.Text Then
        timer1.Enabled = False
        LAccionActual.Text=" "
        BAccion1.Color=Colors.Gray

        BAccion2.Color=Colors.Gray
        BAccion3.Color=Colors.Gray
        BAccion4.Color=Colors.Gray
        BAccion5.Color=Colors.Gray
        BAccion6.Color=Colors.Gray
        BAccion7.Color=Colors.Gray
        BAccion8.Color=Colors.Gray
        BAccion9.Color=Colors.Gray
        BAccion10.Color=Colors.Gray
        BAccion11.Color=Colors.Gray
        BAccion12.Color=Colors.Gray
        contadorcambio=0
        contadortts=0
    End If

End Sub

'Renombra los botones cuando se presiona por un
largo tiempo un boton
Private Sub BAccion1_LongClick
    BAccion1.Text=ETRenombrar.Text
End Sub

Private Sub BAccion2_LongClick
    BAccion2.Text=ETRenombrar.Text
End Sub

Private Sub BAccion3_LongClick
    BAccion3.Text=ETRenombrar.Text
End Sub

Private Sub BAccion4_LongClick
    BAccion4.Text=ETRenombrar.Text
End Sub

Private Sub BAccion5_LongClick
    BAccion5.Text=ETRenombrar.Text
End Sub

Private Sub BAccion6_LongClick
    BAccion6.Text=ETRenombrar.Text
End Sub

Private Sub BAccion7_LongClick
    BAccion7.Text=ETRenombrar.Text
End Sub

Private Sub BAccion8_LongClick
    BAccion8.Text=ETRenombrar.Text

```

```

End Sub

Private Sub BAccion9_LongClick
    BAccion9.Text=ETRenombrar.Text
End Sub

Private Sub BAccion10_LongClick
    BAccion10.Text=ETRenombrar.Text
End Sub

Private Sub BAccion11_LongClick
    BAccion11.Text=ETRenombrar.Text
End Sub

Private Sub BAccion12_LongClick
    BAccion12.Text=ETRenombrar.Text
End Sub

'Reclasifica la accion como no clasificada
Private Sub BCambio_Click
    accion="Unclassified"
    cambio=True

```

```

        BCambio.Color=Colors.Red
End Sub

Private Sub BCambio_LongClick
    cambio=False
    BCambio.Color=Colors.Gray
End Sub

Private Sub BAYuda_Click
    LAccionActual.Set textSizeAnimated(10,10)
    LAccionActual.Text="Tap star to begin.
Long Tap to stop and save a txt file in the internal
memory. If you want to rename an action longclick
any button. The default time for each action is 1
minute."
End Sub

Private Sub BAYuda_LongClick
    LAccionActual.Set textSizeAnimated(10,28)
    LAccionActual.Text="Begin"
End Sub

```

Script para obtener un conjunto de modelos de ML.

Adquisición de los datos.

Se convierten los datos de tabla a tipo numérico para poder procesar las variables

```
RAW_CH_ACX=table2array( RAW_CH(:,1));  
RAW_CH_ACY=table2array( RAW_CH(:,2));  
RAW_CH_ACZ=table2array( RAW_CH(:,3));  
RAW_CH_GYRX=table2array(RAW_CH(:,4));  
RAW_CH_GYRY=table2array(RAW_CH(:,5));  
RAW_CH_GYRZ=table2array(RAW_CH(:,6));  
RAW_CH_LABEL=table2array(RAW_CH(:,7));
```

```
RAW_LA_ACX=table2array( RAW_LA(:,1));  
RAW_LA_ACY=table2array( RAW_LA(:,2));  
RAW_LA_ACZ=table2array( RAW_LA(:,3));  
RAW_LA_GYRX=table2array(RAW_LA(:,4));  
RAW_LA_GYRY=table2array(RAW_LA(:,5));  
RAW_LA_GYRZ=table2array(RAW_LA(:,6));  
RAW_LA_LABEL=table2array(RAW_LA(:,7));
```

```
RAW_RA_ACX=table2array( RAW_RA(:,1));  
RAW_RA_ACY=table2array( RAW_RA(:,2));  
RAW_RA_ACZ=table2array( RAW_RA(:,3));  
RAW_RA_GYRX=table2array(RAW_RA(:,4));  
RAW_RA_GYRY=table2array(RAW_RA(:,5));  
RAW_RA_GYRZ=table2array(RAW_RA(:,6));  
RAW_RA_LABEL=table2array(RAW_RA(:,7));
```

Preprocesamiento

Se configura la frecuencia de actualización de los módulos, los cuales deben de ser homogénea.

```
conf_Facq=100;  
conf_TSpan=1;
```

Se calcula el span a considerar

```
VPRE_Span=fix(conf_Facq*conf_TSpan-1);
```

Se crean los vectores en los cuales se va a almacenar a información preprocesada y se rellenan de ceros.

```
VSize_CH=size(RAW_CH_ACX,1);  
VSize_LA=size(RAW_LA_ACX,1);  
VSize_RA=size(RAW_RA_ACX,1);
```

```
PRE_CH_ACX = zeros(VSize_CH-VPRE_Span,1);  
PRE_CH_ACY = zeros(VSize_CH-VPRE_Span,1);  
PRE_CH_ACZ = zeros(VSize_CH-VPRE_Span,1);  
PRE_CH_GYRX = zeros(VSize_CH-VPRE_Span,1);  
PRE_CH_GYRY = zeros(VSize_CH-VPRE_Span,1);  
PRE_CH_GYRZ = zeros(VSize_CH-VPRE_Span,1);  
PRE_CH_LABEL = zeros(VSize_CH-VPRE_Span,1);
```

```
PRE_LA_ACX = zeros(VSize_LA-VPRE_Span,1);  
PRE_LA_ACY = zeros(VSize_LA-VPRE_Span,1);  
PRE_LA_ACZ = zeros(VSize_LA-VPRE_Span,1);  
PRE_LA_GYRX = zeros(VSize_LA-VPRE_Span,1);  
PRE_LA_GYRY = zeros(VSize_LA-VPRE_Span,1);  
PRE_LA_GYRZ = zeros(VSize_LA-VPRE_Span,1);  
PRE_LA_LABEL = zeros(VSize_LA-VPRE_Span,1);
```

```
PRE_RA_ACX = zeros(VSize_RA-VPRE_Span,1);  
PRE_RA_ACY = zeros(VSize_RA-VPRE_Span,1);  
PRE_RA_ACZ = zeros(VSize_RA-VPRE_Span,1);  
PRE_RA_GYRX = zeros(VSize_RA-VPRE_Span,1);  
PRE_RA_GYRY = zeros(VSize_RA-VPRE_Span,1);  
PRE_RA_GYRZ = zeros(VSize_RA-VPRE_Span,1);  
PRE_RA_LABEL = zeros(VSize_RA-VPRE_Span,1);
```

Se realiza el preprocesamiento de la señal al span requerido.

```
parfor n = 1:(size(RAW_CH,1)-VPRE_Span)
```

```
PRE_CH_ACX(n)=mean(RAW_CH_ACX(n:n+VPRE_Span,1));
```

```
PRE_CH_ACY(n)=mean(RAW_CH_ACY(n:n+VPRE_Span,1));
```

```
PRE_CH_ACZ(n)=mean(RAW_CH_ACZ(n:n+VPRE_Span,1));
```

```
PRE_CH_GYRX(n)=mean(RAW_CH_GYRX(n:n+VPRE_Span,1));
```

```
PRE_CH_GYRY(n)=mean(RAW_CH_GYRY(n:n+VPRE_Span,1));
```

```
PRE_CH_GYRZ(n)=mean(RAW_CH_GYRZ(n:n+VPRE_Span,1));
```

```
end
```

```
PRE_CH_LABEL=RAW_CH_LABEL(1:size(RAW_CH,1)
-VPRE_Span);
PRE_CH=[PRE_CH_ACX, PRE_CH_ACY,
PRE_CH_ACZ, PRE_CH_GYRX, PRE_CH_GYRY,
PRE_CH_GYRZ, PRE_CH_LABEL];
```

```
parfor n = 1:(size(RAW_LA,1)-VPRE_Span)
```

```
PRE_LA_ACX(n)=mean(RAW_LA_ACX(n:n+VPRE_Sp
an,1));
```

```
PRE_LA_ACY(n)=mean(RAW_LA_ACY(n:n+VPRE_Sp
an,1));
```

```
PRE_LA_ACZ(n)=mean(RAW_LA_ACZ(n:n+VPRE_Sp
an,1));
```

```
PRE_LA_GYRX(n)=mean(RAW_LA_GYRX(n:n+VPRE_
Span,1));
```

```
PRE_LA_GYRY(n)=mean(RAW_LA_GYRY(n:n+VPRE_
Span,1));
```

```
PRE_LA_GYRZ(n)=mean(RAW_LA_GYRZ(n:n+VPRE_
Span,1));
```

```
end
```

```
PRE_LA_LABEL=RAW_LA_LABEL(1:size(RAW_LA,1)-
VPRE_Span);
```

```
PRE_LA=[PRE_LA_ACX, PRE_LA_ACY, PRE_LA_ACZ,
PRE_LA_GYRX, PRE_LA_GYRY, PRE_LA_GYRZ,
PRE_LA_LABEL];
```

```
parfor n = 1:(size(RAW_RA,1)-VPRE_Span)
```

```
PRE_RA_ACX(n)=mean(RAW_RA_ACX(n:n+VPRE_Sp
an,1));
```

```
PRE_RA_ACY(n)=mean(RAW_RA_ACY(n:n+VPRE_Sp
an,1));
```

```
PRE_RA_ACZ(n)=mean(RAW_RA_ACZ(n:n+VPRE_Sp
an,1));
```

```
PRE_RA_GYRX(n)=mean(RAW_RA_GYRX(n:n+VPRE_
_Span,1));
```

```
PRE_RA_GYRY(n)=mean(RAW_RA_GYRY(n:n+VPRE_
_Span,1));
```

```
PRE_RA_GYRZ(n)=mean(RAW_RA_GYRZ(n:n+VPRE_
Span,1));
end
```

```
PRE_RA_LABEL=RAW_RA_LABEL(1:size(RAW_RA,1)
-VPRE_Span);
PRE_RA=[PRE_RA_ACX, PRE_RA_ACY, PRE_RA_ACZ,
PRE_RA_GYRX, PRE_RA_GYRY, PRE_RA_GYRZ,
PRE_RA_LABEL];
```

%Mantenimiento: Limpiar variables ya
descuapadas

```
files1={'PRE_CH_ACX', 'PRE_CH_ACY',
'PRE_CH_ACZ', 'PRE_CH_GYRX', 'PRE_CH_GYRY',
'PRE_CH_GYRZ', 'PRE_CH_LABEL', 'PRE_LA_ACX',
'PRE_LA_ACY', 'PRE_LA_ACZ', 'PRE_LA_GYRX',
PRE_LA_GYRY, PRE_LA_GYRZ, PRE_LA_LABEL,
PRE_RA_ACX, PRE_RA_ACY, PRE_RA_ACZ,
PRE_RA_GYRX, PRE_RA_GYRY, PRE_RA_GYRZ,
PRE_RA_LABEL', 'RAW_CH_ACX', 'RAW_CH_ACY',
'RAW_CH_ACZ', 'RAW_CH_GYRX', 'RAW_CH_GYRY',
'RAW_CH_GYRZ', 'RAW_LA_ACX', 'RAW_LA_ACY',
'RAW_LA_ACZ', 'RAW_LA_GYRX', 'RAW_LA_GYRY',
'RAW_LA_GYRZ', 'RAW_RA_ACX', 'RAW_RA_ACY',
'RAW_RA_ACZ', 'RAW_RA_GYRX', 'RAW_RA_GYRY',
'RAW_RA_GYRZ', 'VSize_CH', 'VSize_LA',
'VSize_RA'};
```

```
files2={'PRE_CH_ACX', 'PRE_CH_ACY',
'PRE_CH_ACZ', 'PRE_CH_GYRX', 'PRE_CH_GYRY',
'PRE_CH_GYRZ', 'PRE_CH_LABEL', 'PRE_LA_ACX',
'PRE_LA_ACY', 'PRE_LA_ACZ', 'PRE_LA_GYRX',
'PRE_LA_GYRY', 'PRE_LA_GYRZ', 'PRE_LA_LABEL',
'PRE_RA_ACX', 'PRE_RA_ACY', 'PRE_RA_ACZ',
'PRE_RA_GYRX', 'PRE_RA_GYRY', 'PRE_RA_GYRZ',
'PRE_RA_LABEL'};
```

```
files3={'RAW_CH_LABEL', 'RAW_LA_LABEL',
'RAW_RA_LABEL', 'VPRE_Span', 'conf_Facq',
'conf_TSpan'};
clear (files1{:});
clear (files2{:});
clear (files3{:});
```

Obtención de características estadísticas

Se obtienen 14 características estadísticas a partir
de las señales preprocesadas.

```
conf_Facq=100;
conf_TSpan=2;
```

Se calcula el span a considerar
VEST_Span=fix(conf_Facq*conf_TSpan-1);

Se crean los vectores en los cuales se va a
almacenar a información preprcesada y se rellenan
de ceros.

```
VSize_CH=size(PRE_CH,1);
VSize_LA=size(PRE_LA,1);
VSize_RA=size(PRE_RA,1);
```

```
SSF_CH= zeros(VSize_CH-VEST_Span,85);
SSF_LA= zeros(VSize_LA-VEST_Span,85);
SSF_RA= zeros(VSize_RA-VEST_Span,85);
```

Se realiza el preprocesamiento de la señal al span requerido.

```
for n = 1:(VSize_CH-VEST_Span)
```

```
    SSF_CH(n,1)=mean(PRE_CH(n:n+VEST_Span,1));
```

```
    SSF_CH(n,2)=mean(PRE_CH(n:n+VEST_Span,2));
```

```
    SSF_CH(n,3)=mean(PRE_CH(n:n+VEST_Span,3));
```

```
    SSF_CH(n,4)=mean(PRE_CH(n:n+VEST_Span,4));
```

```
    SSF_CH(n,5)=mean(PRE_CH(n:n+VEST_Span,5));
```

```
    SSF_CH(n,6)=mean(PRE_CH(n:n+VEST_Span,6));
```

```
    SSF_CH(n,7)=harmmean(PRE_CH(n:n+VEST_Span,1
));
```

```
    SSF_CH(n,8)=harmmean(PRE_CH(n:n+VEST_Span,2
));
```

```
    SSF_CH(n,9)=harmmean(PRE_CH(n:n+VEST_Span,3
));
```

```
    SSF_CH(n,10)=harmmean(PRE_CH(n:n+VEST_Span,
4));
```

```
    SSF_CH(n,11)=harmmean(PRE_CH(n:n+VEST_Span,
5));
```

```
    SSF_CH(n,12)=harmmean(PRE_CH(n:n+VEST_Span,
6));
```

```
    SSF_CH(n,13)=kurtosis(PRE_CH(n:n+VEST_Span,1));
```

```
    SSF_CH(n,14)=kurtosis(PRE_CH(n:n+VEST_Span,2));
```

```
    SSF_CH(n,15)=kurtosis(PRE_CH(n:n+VEST_Span,3));
```

```
    SSF_CH(n,16)=kurtosis(PRE_CH(n:n+VEST_Span,4));
```

```
    SSF_CH(n,17)=kurtosis(PRE_CH(n:n+VEST_Span,5));
```

```
    SSF_CH(n,18)=kurtosis(PRE_CH(n:n+VEST_Span,6));
```

```
    SSF_CH(n,19)=median(PRE_CH(n:n+VEST_Span,1));
```

```
    SSF_CH(n,20)=median(PRE_CH(n:n+VEST_Span,2));
```

```
    SSF_CH(n,21)=median(PRE_CH(n:n+VEST_Span,3));
```

```
    SSF_CH(n,22)=median(PRE_CH(n:n+VEST_Span,4));
```

```
    SSF_CH(n,23)=median(PRE_CH(n:n+VEST_Span,5));
```

```
    SSF_CH(n,24)=median(PRE_CH(n:n+VEST_Span,6));
```

```
    SSF_CH(n,25)=mode(PRE_CH(n:n+VEST_Span,1));
```

```
    SSF_CH(n,26)=mode(PRE_CH(n:n+VEST_Span,2));
```

```
    SSF_CH(n,27)=mode(PRE_CH(n:n+VEST_Span,3));
```

```
    SSF_CH(n,28)=mode(PRE_CH(n:n+VEST_Span,4));
```

```
    SSF_CH(n,29)=mode(PRE_CH(n:n+VEST_Span,5));
```

```
    SSF_CH(n,30)=mode(PRE_CH(n:n+VEST_Span,6));
```

```
        SSF_CH(n,31)=min(PRE_CH(n:n+VEST_Span,1));
```

```
        SSF_CH(n,32)=min(PRE_CH(n:n+VEST_Span,2));
```

```
        SSF_CH(n,33)=min(PRE_CH(n:n+VEST_Span,3));
```

```
        SSF_CH(n,34)=min(PRE_CH(n:n+VEST_Span,4));
```

```
        SSF_CH(n,35)=min(PRE_CH(n:n+VEST_Span,5));
```

```
        SSF_CH(n,36)=min(PRE_CH(n:n+VEST_Span,6));
```

```
    SSF_CH(n,37)=max(PRE_CH(n:n+VEST_Span,1));
```

```
    SSF_CH(n,38)=max(PRE_CH(n:n+VEST_Span,2));
```

```
    SSF_CH(n,39)=max(PRE_CH(n:n+VEST_Span,3));
```

```
    SSF_CH(n,40)=max(PRE_CH(n:n+VEST_Span,4));
```

```
    SSF_CH(n,41)=max(PRE_CH(n:n+VEST_Span,5));
```

```
    SSF_CH(n,42)=max(PRE_CH(n:n+VEST_Span,6));
```

```

SSF_CH(n,43)=peak2peak(PRE_CH(n:n+VEST_Span,
1));

SSF_CH(n,44)=peak2peak(PRE_CH(n:n+VEST_Span,
2));

SSF_CH(n,45)=peak2peak(PRE_CH(n:n+VEST_Span,
3));

SSF_CH(n,46)=peak2peak(PRE_CH(n:n+VEST_Span,
4));

SSF_CH(n,47)=peak2peak(PRE_CH(n:n+VEST_Span,
5));

SSF_CH(n,48)=peak2peak(PRE_CH(n:n+VEST_Span,
6));

    SSF_CH(n,49)=rms(PRE_CH(n:n+VEST_Span,1));
    SSF_CH(n,50)=rms(PRE_CH(n:n+VEST_Span,2));
    SSF_CH(n,51)=rms(PRE_CH(n:n+VEST_Span,3));
    SSF_CH(n,52)=rms(PRE_CH(n:n+VEST_Span,4));
    SSF_CH(n,53)=rms(PRE_CH(n:n+VEST_Span,5));
    SSF_CH(n,54)=rms(PRE_CH(n:n+VEST_Span,6));

SSF_CH(n,55)=skewness(PRE_CH(n:n+VEST_Span,1)
);

SSF_CH(n,56)=skewness(PRE_CH(n:n+VEST_Span,2)
);

SSF_CH(n,57)=skewness(PRE_CH(n:n+VEST_Span,3)
);

SSF_CH(n,58)=skewness(PRE_CH(n:n+VEST_Span,4)
);

SSF_CH(n,59)=skewness(PRE_CH(n:n+VEST_Span,5)
);

SSF_CH(n,60)=skewness(PRE_CH(n:n+VEST_Span,6)
);

    SSF_CH(n,61)=std(PRE_CH(n:n+VEST_Span,1));
    SSF_CH(n,62)=std(PRE_CH(n:n+VEST_Span,2));
    SSF_CH(n,63)=std(PRE_CH(n:n+VEST_Span,3));
    SSF_CH(n,64)=std(PRE_CH(n:n+VEST_Span,4));
    SSF_CH(n,65)=std(PRE_CH(n:n+VEST_Span,5));
    SSF_CH(n,66)=std(PRE_CH(n:n+VEST_Span,6));

SSF_CH(n,67)=sum(PRE_CH(n:n+VEST_Span,1));

SSF_CH(n,68)=sum(PRE_CH(n:n+VEST_Span,2));

SSF_CH(n,69)=sum(PRE_CH(n:n+VEST_Span,3));

SSF_CH(n,70)=sum(PRE_CH(n:n+VEST_Span,4));

SSF_CH(n,71)=sum(PRE_CH(n:n+VEST_Span,5));

SSF_CH(n,72)=sum(PRE_CH(n:n+VEST_Span,6));

SSF_CH(n,73)=trimmean(PRE_CH(n:n+VEST_Span,1
),10);

SSF_CH(n,74)=trimmean(PRE_CH(n:n+VEST_Span,2
),10);

SSF_CH(n,75)=trimmean(PRE_CH(n:n+VEST_Span,3
),10);

SSF_CH(n,76)=trimmean(PRE_CH(n:n+VEST_Span,4
),10);

SSF_CH(n,77)=trimmean(PRE_CH(n:n+VEST_Span,5
),10);

SSF_CH(n,78)=trimmean(PRE_CH(n:n+VEST_Span,6
),10);

    SSF_CH(n,79)=var(PRE_CH(n:n+VEST_Span,1));
    SSF_CH(n,80)=var(PRE_CH(n:n+VEST_Span,2));
    SSF_CH(n,81)=var(PRE_CH(n:n+VEST_Span,3));
    SSF_CH(n,82)=var(PRE_CH(n:n+VEST_Span,4));
    SSF_CH(n,83)=var(PRE_CH(n:n+VEST_Span,5));
    SSF_CH(n,84)=var(PRE_CH(n:n+VEST_Span,6));

SSF_CH(n,85)=PRE_CH(n,7);
end

for n = 1:(VSize_LA-VEST_Span)

    SSF_LA(n,1)=mean(PRE_LA(n:n+VEST_Span,1));
    SSF_LA(n,2)=mean(PRE_LA(n:n+VEST_Span,2));
    SSF_LA(n,3)=mean(PRE_LA(n:n+VEST_Span,3));
    SSF_LA(n,4)=mean(PRE_LA(n:n+VEST_Span,4));
    SSF_LA(n,5)=mean(PRE_LA(n:n+VEST_Span,5));
    SSF_LA(n,6)=mean(PRE_LA(n:n+VEST_Span,6));

```

```

SSF_LA(n,7)=harmmean(PRE_LA(n:n+VEST_Span,1))
;

SSF_LA(n,8)=harmmean(PRE_LA(n:n+VEST_Span,2))
;

SSF_LA(n,9)=harmmean(PRE_LA(n:n+VEST_Span,3))
;

SSF_LA(n,10)=harmmean(PRE_LA(n:n+VEST_Span,4
));

SSF_LA(n,11)=harmmean(PRE_LA(n:n+VEST_Span,5
));

SSF_LA(n,12)=harmmean(PRE_LA(n:n+VEST_Span,6
));

SSF_LA(n,13)=kurtosis(PRE_LA(n:n+VEST_Span,1));

SSF_LA(n,14)=kurtosis(PRE_LA(n:n+VEST_Span,2));

SSF_LA(n,15)=kurtosis(PRE_LA(n:n+VEST_Span,3));

SSF_LA(n,16)=kurtosis(PRE_LA(n:n+VEST_Span,4));

SSF_LA(n,17)=kurtosis(PRE_LA(n:n+VEST_Span,5));

SSF_LA(n,18)=kurtosis(PRE_LA(n:n+VEST_Span,6));

SSF_LA(n,19)=median(PRE_LA(n:n+VEST_Span,1));

SSF_LA(n,20)=median(PRE_LA(n:n+VEST_Span,2));

SSF_LA(n,21)=median(PRE_LA(n:n+VEST_Span,3));

SSF_LA(n,22)=median(PRE_LA(n:n+VEST_Span,4));

SSF_LA(n,23)=median(PRE_LA(n:n+VEST_Span,5));

SSF_LA(n,24)=median(PRE_LA(n:n+VEST_Span,6));

SSF_LA(n,25)=mode(PRE_LA(n:n+VEST_Span,1));

SSF_LA(n,26)=mode(PRE_LA(n:n+VEST_Span,2));

SSF_LA(n,27)=mode(PRE_LA(n:n+VEST_Span,3));

```

```

SSF_LA(n,28)=mode(PRE_LA(n:n+VEST_Span,4));

SSF_LA(n,29)=mode(PRE_LA(n:n+VEST_Span,5));

SSF_LA(n,30)=mode(PRE_LA(n:n+VEST_Span,6));

SSF_LA(n,31)=min(PRE_LA(n:n+VEST_Span,1));
SSF_LA(n,32)=min(PRE_LA(n:n+VEST_Span,2));
SSF_LA(n,33)=min(PRE_LA(n:n+VEST_Span,3));
SSF_LA(n,34)=min(PRE_LA(n:n+VEST_Span,4));
SSF_LA(n,35)=min(PRE_LA(n:n+VEST_Span,5));
SSF_LA(n,36)=min(PRE_LA(n:n+VEST_Span,6));

SSF_LA(n,37)=max(PRE_LA(n:n+VEST_Span,1));
SSF_LA(n,38)=max(PRE_LA(n:n+VEST_Span,2));
SSF_LA(n,39)=max(PRE_LA(n:n+VEST_Span,3));
SSF_LA(n,40)=max(PRE_LA(n:n+VEST_Span,4));
SSF_LA(n,41)=max(PRE_LA(n:n+VEST_Span,5));
SSF_LA(n,42)=max(PRE_LA(n:n+VEST_Span,6));

SSF_LA(n,43)=peak2peak(PRE_LA(n:n+VEST_Span,1
));

SSF_LA(n,44)=peak2peak(PRE_LA(n:n+VEST_Span,2
));

SSF_LA(n,45)=peak2peak(PRE_LA(n:n+VEST_Span,3
));

SSF_LA(n,46)=peak2peak(PRE_LA(n:n+VEST_Span,4
));

SSF_LA(n,47)=peak2peak(PRE_LA(n:n+VEST_Span,5
));

SSF_LA(n,48)=peak2peak(PRE_LA(n:n+VEST_Span,6
));

SSF_LA(n,49)=rms(PRE_LA(n:n+VEST_Span,1));
SSF_LA(n,50)=rms(PRE_LA(n:n+VEST_Span,2));
SSF_LA(n,51)=rms(PRE_LA(n:n+VEST_Span,3));
SSF_LA(n,52)=rms(PRE_LA(n:n+VEST_Span,4));
SSF_LA(n,53)=rms(PRE_LA(n:n+VEST_Span,5));
SSF_LA(n,54)=rms(PRE_LA(n:n+VEST_Span,6));

SSF_LA(n,55)=skewness(PRE_LA(n:n+VEST_Span,1))
;

```

```

SSF_LA(n,56)=skewness(PRE_LA(n:n+VEST_Span,2))
;

SSF_LA(n,57)=skewness(PRE_LA(n:n+VEST_Span,3))
;

SSF_LA(n,58)=skewness(PRE_LA(n:n+VEST_Span,4))
;

SSF_LA(n,59)=skewness(PRE_LA(n:n+VEST_Span,5))
;

SSF_LA(n,60)=skewness(PRE_LA(n:n+VEST_Span,6))
;

    SSF_LA(n,61)=std(PRE_LA(n:n+VEST_Span,1));
    SSF_LA(n,62)=std(PRE_LA(n:n+VEST_Span,2));
    SSF_LA(n,63)=std(PRE_LA(n:n+VEST_Span,3));
    SSF_LA(n,64)=std(PRE_LA(n:n+VEST_Span,4));
    SSF_LA(n,65)=std(PRE_LA(n:n+VEST_Span,5));
    SSF_LA(n,66)=std(PRE_LA(n:n+VEST_Span,6));

    SSF_LA(n,67)=sum(PRE_LA(n:n+VEST_Span,1));
    SSF_LA(n,68)=sum(PRE_LA(n:n+VEST_Span,2));
    SSF_LA(n,69)=sum(PRE_LA(n:n+VEST_Span,3));
    SSF_LA(n,70)=sum(PRE_LA(n:n+VEST_Span,4));
    SSF_LA(n,71)=sum(PRE_LA(n:n+VEST_Span,5));
    SSF_LA(n,72)=sum(PRE_LA(n:n+VEST_Span,6));

SSF_LA(n,73)=trimmean(PRE_LA(n:n+VEST_Span,1)
,10);

SSF_LA(n,74)=trimmean(PRE_LA(n:n+VEST_Span,2)
,10);

SSF_LA(n,75)=trimmean(PRE_LA(n:n+VEST_Span,3)
,10);

SSF_LA(n,76)=trimmean(PRE_LA(n:n+VEST_Span,4)
,10);

SSF_LA(n,77)=trimmean(PRE_LA(n:n+VEST_Span,5)
,10);

SSF_LA(n,78)=trimmean(PRE_LA(n:n+VEST_Span,6)
,10);

    SSF_LA(n,79)=var(PRE_LA(n:n+VEST_Span,1));
    SSF_LA(n,80)=var(PRE_LA(n:n+VEST_Span,2));
    SSF_LA(n,81)=var(PRE_LA(n:n+VEST_Span,3));

    SSF_LA(n,82)=var(PRE_LA(n:n+VEST_Span,4));
    SSF_LA(n,83)=var(PRE_LA(n:n+VEST_Span,5));
    SSF_LA(n,84)=var(PRE_LA(n:n+VEST_Span,6));

    SSF_LA(n,85)=PRE_LA(n,7);
end

for n = 1:(VSize_RA-VEST_Span)

    SSF_RA(n,1)=mean(PRE_RA(n:n+VEST_Span,1));

    SSF_RA(n,2)=mean(PRE_RA(n:n+VEST_Span,2));

    SSF_RA(n,3)=mean(PRE_RA(n:n+VEST_Span,3));

    SSF_RA(n,4)=mean(PRE_RA(n:n+VEST_Span,4));

    SSF_RA(n,5)=mean(PRE_RA(n:n+VEST_Span,5));

    SSF_RA(n,6)=mean(PRE_RA(n:n+VEST_Span,6));

    SSF_RA(n,7)=harmmean(PRE_RA(n:n+VEST_Span,1)
);

    SSF_RA(n,8)=harmmean(PRE_RA(n:n+VEST_Span,2)
);

    SSF_RA(n,9)=harmmean(PRE_RA(n:n+VEST_Span,3)
);

    SSF_RA(n,10)=harmmean(PRE_RA(n:n+VEST_Span,
4));

    SSF_RA(n,11)=harmmean(PRE_RA(n:n+VEST_Span,
5));

    SSF_RA(n,12)=harmmean(PRE_RA(n:n+VEST_Span,
6));

    SSF_RA(n,13)=kurtosis(PRE_RA(n:n+VEST_Span,1));

    SSF_RA(n,14)=kurtosis(PRE_RA(n:n+VEST_Span,2));

    SSF_RA(n,15)=kurtosis(PRE_RA(n:n+VEST_Span,3));

    SSF_RA(n,16)=kurtosis(PRE_RA(n:n+VEST_Span,4));

    SSF_RA(n,17)=kurtosis(PRE_RA(n:n+VEST_Span,5));

```

```

SSF_RA(n,18)=kurtosis(PRE_RA(n:n+VEST_Span,6));

SSF_RA(n,19)=median(PRE_RA(n:n+VEST_Span,1));
SSF_RA(n,20)=median(PRE_RA(n:n+VEST_Span,2));
SSF_RA(n,21)=median(PRE_RA(n:n+VEST_Span,3));
SSF_RA(n,22)=median(PRE_RA(n:n+VEST_Span,4));
SSF_RA(n,23)=median(PRE_RA(n:n+VEST_Span,5));
SSF_RA(n,24)=median(PRE_RA(n:n+VEST_Span,6));

SSF_RA(n,25)=mode(PRE_RA(n:n+VEST_Span,1));
SSF_RA(n,26)=mode(PRE_RA(n:n+VEST_Span,2));
SSF_RA(n,27)=mode(PRE_RA(n:n+VEST_Span,3));
SSF_RA(n,28)=mode(PRE_RA(n:n+VEST_Span,4));
SSF_RA(n,29)=mode(PRE_RA(n:n+VEST_Span,5));
SSF_RA(n,30)=mode(PRE_RA(n:n+VEST_Span,6));

    SSF_RA(n,31)=min(PRE_RA(n:n+VEST_Span,1));
    SSF_RA(n,32)=min(PRE_RA(n:n+VEST_Span,2));
    SSF_RA(n,33)=min(PRE_RA(n:n+VEST_Span,3));
    SSF_RA(n,34)=min(PRE_RA(n:n+VEST_Span,4));
    SSF_RA(n,35)=min(PRE_RA(n:n+VEST_Span,5));
    SSF_RA(n,36)=min(PRE_RA(n:n+VEST_Span,6));

SSF_RA(n,37)=max(PRE_RA(n:n+VEST_Span,1));
SSF_RA(n,38)=max(PRE_RA(n:n+VEST_Span,2));
SSF_RA(n,39)=max(PRE_RA(n:n+VEST_Span,3));
SSF_RA(n,40)=max(PRE_RA(n:n+VEST_Span,4));
SSF_RA(n,41)=max(PRE_RA(n:n+VEST_Span,5));
SSF_RA(n,42)=max(PRE_RA(n:n+VEST_Span,6));

SSF_RA(n,43)=peak2peak(PRE_RA(n:n+VEST_Span,
1));

SSF_RA(n,44)=peak2peak(PRE_RA(n:n+VEST_Span,
2));
SSF_RA(n,45)=peak2peak(PRE_RA(n:n+VEST_Span,
3));
SSF_RA(n,46)=peak2peak(PRE_RA(n:n+VEST_Span,
4));
SSF_RA(n,47)=peak2peak(PRE_RA(n:n+VEST_Span,
5));
SSF_RA(n,48)=peak2peak(PRE_RA(n:n+VEST_Span,
6));

    SSF_RA(n,49)=rms(PRE_RA(n:n+VEST_Span,1));
    SSF_RA(n,50)=rms(PRE_RA(n:n+VEST_Span,2));
    SSF_RA(n,51)=rms(PRE_RA(n:n+VEST_Span,3));
    SSF_RA(n,52)=rms(PRE_RA(n:n+VEST_Span,4));
    SSF_RA(n,53)=rms(PRE_RA(n:n+VEST_Span,5));
    SSF_RA(n,54)=rms(PRE_RA(n:n+VEST_Span,6));

SSF_RA(n,55)=skewness(PRE_RA(n:n+VEST_Span,1)
);
SSF_RA(n,56)=skewness(PRE_RA(n:n+VEST_Span,2)
);
SSF_RA(n,57)=skewness(PRE_RA(n:n+VEST_Span,3)
);
SSF_RA(n,58)=skewness(PRE_RA(n:n+VEST_Span,4)
);
SSF_RA(n,59)=skewness(PRE_RA(n:n+VEST_Span,5)
);
SSF_RA(n,60)=skewness(PRE_RA(n:n+VEST_Span,6)
);

    SSF_RA(n,61)=std(PRE_RA(n:n+VEST_Span,1));
    SSF_RA(n,62)=std(PRE_RA(n:n+VEST_Span,2));
    SSF_RA(n,63)=std(PRE_RA(n:n+VEST_Span,3));
    SSF_RA(n,64)=std(PRE_RA(n:n+VEST_Span,4));
    SSF_RA(n,65)=std(PRE_RA(n:n+VEST_Span,5));
    SSF_RA(n,66)=std(PRE_RA(n:n+VEST_Span,6));

SSF_RA(n,67)=sum(PRE_RA(n:n+VEST_Span,1));

```

```
SSF_RA(n,68)=sum(PRE_RA(n:n+VEST_Span,2));
```

```
SSF_RA(n,69)=sum(PRE_RA(n:n+VEST_Span,3));
```

```
SSF_RA(n,70)=sum(PRE_RA(n:n+VEST_Span,4));
```

```
SSF_RA(n,71)=sum(PRE_RA(n:n+VEST_Span,5));
```

```
SSF_RA(n,72)=sum(PRE_RA(n:n+VEST_Span,6));
```

```
SSF_RA(n,73)=trimmean(PRE_RA(n:n+VEST_Span,1),10);
```

```
SSF_RA(n,74)=trimmean(PRE_RA(n:n+VEST_Span,2),10);
```

```
SSF_RA(n,75)=trimmean(PRE_RA(n:n+VEST_Span,3),10);
```

```
SSF_RA(n,76)=trimmean(PRE_RA(n:n+VEST_Span,4),10);
```

```
SSF_RA(n,77)=trimmean(PRE_RA(n:n+VEST_Span,5),10);
```

```
SSF_RA(n,78)=trimmean(PRE_RA(n:n+VEST_Span,6),10);
```

```
SSF_RA(n,79)=var(PRE_RA(n:n+VEST_Span,1));
```

```
SSF_RA(n,80)=var(PRE_RA(n:n+VEST_Span,2));
```

```
SSF_RA(n,81)=var(PRE_RA(n:n+VEST_Span,3));
```

```
SSF_RA(n,82)=var(PRE_RA(n:n+VEST_Span,4));
```

```
SSF_RA(n,83)=var(PRE_RA(n:n+VEST_Span,5));
```

```
SSF_RA(n,84)=var(PRE_RA(n:n+VEST_Span,6));
```

```
SSF_RA(n,85)=PRE_RA(n,7);
```

```
end
```

```
%Mantenimiento: Limpiar variables ya  
descocupadas
```

```
files={'conf_Facq', 'conf_TSpan', 'files','files1',  
'files2', 'files3', 'n', 'VEST_Span', 'VSize_CH',  
'VSize_LA', 'VSize_RA'};  
clear (files{:});
```

```
Ver modelo CH
```

```
view(DTC_DAGO.ClassificationTree,'Mode','graph')
```

```
view(DTM_DAGO.ClassificationTree,'Mode','graph')
```

```
view(DTF_DAGO.ClassificationTree,'Mode','graph')
```

