

Implementación de técnicas de *Machine Learning* para el diseño de un mecanismo exoesqueleto controlado por señales EMG

JORGE IVAN PALACIOS PARDO

Universidad Santo Tomás
Facultad de Ingeniería Electrónica
Bogotá, Colombia
2024



Implementación de técnicas de *Machine Learning* para el diseño de un mecanismo exoesqueleto controlado por señales EMG

JORGE IVAN PALACIOS PARDO

Proyecto de grado presentado como requisito para optar al título de:

Ingeniero Electrónico

Director:

David Alejandro Martínez Vásquez Ph.D

Codirector:

Gianina Garrido Silva M.Sc

Grupo de Investigación GED
Línea de Investigación:
Universidad Santo Tomás
Facultad de Ingeniería Electrónica
Bogotá, Colombia
2024

Autoridades de la universidad

RECTOR GENERAL

R.P. FRAY ÁLVARO JOSÉ ARANGO RESTREPO, O.P.

VICERRECTOR ADMINISTRATIVO Y FINANCIERO GENERAL

R.P. FRAY HERNÁN YESID RIVERA ROBERTO, O.P.

VICERRECTOR ACADÉMICO GENERAL

R.P. FRAY MAURICIO ANTONIO CORTÉS GALLEGUO, O.P.

SECRETARIO GENERAL

Dra. INGRID LORENA CAMPOS VARGAS

DECANO DIVISIÓN DE INGENIERÍAS

R.P. FRAY JAVIER ANTONIO HINCAPIÉ ARDILA, O.P.

SECRETARIA DE DIVISIÓN

E.C. LUZ PATRICIA ROCHA CAICEDO

DECANO FACULTAD DE INGENIERÍA ELECTRÓNICA

Ph.D CARLOS ANDRÉS TORRES PINZÓN

Nota de aceptación

Firma del tutor

Firma del jurado

Firma del jurado

BOGOTÁ D.C. _____ DE 2024

ADVERTENCIA

La Universidad Santo Tomás no se hace responsable de las opiniones y conceptos expresados en el trabajo de grado, solo velará por qué no se publique nada contrario al dogma ni a la moral católica y porque el trabajo no tenga ataques personales y únicamente se vea el anhelo de buscar la verdad científica.

Capítulo III –Art. 46 del Reglamento de la Universidad Santo Tomás.

Contenido

Resumen	VIII
Abstract.....	IX
Lista de Figuras.....	X
Lista de Tablas	XI
Glosario.....	XII
1. Introducción	1
1.1. Planteamiento del Problema	2
1.2. Objetivos	4
1.2.1. Objetivo General	4
1.2.2. Objetivos Específicos.....	4
1.3. Justificación.....	5
1.4. Impacto Social.....	6
2. Estado del Arte	7
2.1. Procesamiento y análisis de las señales EMG.....	7
2.2. Robótica aplicada.....	8
2.3. Estudio de Enfermedades	9
3. Marco Teórico.....	10
3.1. Señales de Electromiografía	10
3.2. Algoritmos:	10
3.2.1. Algoritmos de Aprendizaje por Refuerzo:	10
3.2.2. Algoritmos no Supervisados:	12
3.2.3. Algoritmos Supervisados:	13
3.2.3.1. Algoritmo KNN.....	14
3.2.3.2. Algoritmo Random Forest.....	16
3.2.3.3. Algoritmo Naïve Bayes.....	17
3.2.3.4. Algoritmo SVM	18
3.2.3.5. Algoritmo MaxVoting:	19
4. Diseño Metodológico	21
4.1. Fases y Metodología.....	21
4.1.1 Búsqueda Bibliográfica	21
5. Desarrollo Conceptual	22
5.1. Selección del <i>Dataset</i>	22

5.2. Algoritmos Clasificatorios	23
5.2.1. Algoritmo KNN	25
5.2.2. Algoritmo Random Forest	26
5.2.3. Algoritmo Naïve Bayes	26
5.2.4. Algoritmo SVM	27
5.2.5. Algoritmo MaxVoting	28
6. Análisis de Resultados y Discusión	28
6.1. Análisis Resultados de los Algoritmos.....	28
6.2. Diseño Estrategia de Control del Exoesqueleto	34
6.2.1. Selección Microcontrolador e Implementación de Software	35
6.2.2. Elementos de Hardware del Exoesqueleto	36
6.2.3. Propuesta de la Estrategia de Control	38
7. Conclusiones y Trabajos futuros	46
8. Referencias Bibliográficas	47

Resumen

El campo industrial requiere que sus trabajadores efectúen actividad física pesada, una de las actividades más comunes es el levantamiento y carga de objetos pesados, la cual requiere de gran activación muscular en los bíceps dependiendo del peso del objeto. El desarrollo incorrecto de dicha actividad puede resultar en heridas musculoesqueléticas como desgarres musculares y desarrollo de tendinitis por lo que es importante el desarrollo en forma adecuada de estas. Para ello se puede implementar el machine learning y datasets ya existentes de las señales de activación muscular (Electromiografía) de los bíceps para determinar cuando una persona hace un levantamiento erróneo y así prevenir una lesión muscular. Esto se desarrollará con la creación de un algoritmo clasificador mediante el lenguaje de programación Python y una base de datos de un trabajo previo con finalidad similar y el resultado de estos conlleva al desarrollo teórico de un exoesqueleto para el brazo que pueda reducir la tensión muscular del usuario cuando efectúe un movimiento que ponga en riesgo la integridad de los bíceps.

Palabras clave: Electromiografía, activación muscular, lesiones musculoesqueléticas, machine learning, dataset, algoritmo clasificador, bíceps, exoesqueleto.

Abstract

The industrial field requires that its workers effectuate heavy physical activity, one of the most common activities is the lifting and carrying of heavy objects, which requires great muscle activation on the biceps, depending on how heavy the object is. The incorrect development of such activity can result in skeletal-muscle injuries like muscle tears and the development of tendinitis, so it is important to develop these activities adequately. For this reason, machine learning and existing datasets of electromyography can be implemented to determine when a person is doing an incorrect lifting and thus prevent a muscular injury. This will be developed with the creation of a classifier algorithm through the programming language Python and a dataset from a previous study with a similar purpose, and the result will lead to the theoretical development of an arm exoskeleton that could reduce muscle tension of the user when they effectuate a movement that could risk the bicep's integrity.

Keywords: Electromyography, muscle activation, skeletal-muscle injury, machine learning, dataset, classifier algorithm, bicep, exoskeleton.

Lista de Figuras

Figura 3-1 Funcionamiento algoritmo KNN [26]	14
Figura 3-2 Diagrama de frontera de decisión KNN [26]	15
Figura 3-3 Diagrama de Arboles Random Forest [27]	16
Figura 3-4 Funcionamiento Algoritmo SVM [31]	18
Figura 5-1 Gráfica ROC de los algoritmos	23
Figura 5-2 Gráfica de dispersión del Dataset	24
Figura 5-3 Boxplot del Dataset	25
Figura 6-1 Gráfica de barras de exactitud y precisión de los algoritmos	29
Figura 6-2 Matrices de confusión Random Forest y MaxVoting	29
Figura 6-3 Diagramas de frontera de decisión Random Forest y MaxVoting	30
Figura 6-4 Diagramas de frontera de decisión Random Forest y MaxVoting con Zoom	30
Figura 6-5 Matrices de confusión KNN	31
Figura 6-6 Diagramas de frontera de decisión KNN	31
Figura 6-7 Diagramas de frontera de decisión KNN con Zoom	32
Figura 6-8 Matriz de Confusión SVM	32
Figura 6-9 Diagrama de frontera de decisión SVM	33
Figura 6-10 Matriz de Confusión Naïve Bayes	33
Figura 6-11 Diagrama de frontera de decisión Naïve Bayes	34
Figura 6-12 Montaje de la toma de datos [33]	35
Figura 6-13 Ubicación de las cabezas de los bíceps [35]	36
Figura 6-14 Posicionamiento Sensor EMG [36]	37
Figura 6-15 Dibujo Exoesqueleto vista frontal y trasera desarrollada en Krita	38
Figura 6-16 Diagrama del Funcionamiento del Algoritmo	38
Figura 6-17 Diagrama del Funcionamiento del Exoesqueleto	39
Figura 6-18 Movimientos que causan más esfuerzo en las cabezas de los bíceps	40
Figura 6-19 Diagrama del Bloques del Control Proporcional de Lazo Cerrado	40
Figura 6-20 Diagrama de Flujo del Código	42
Figura 6-21 Simulación del Control Proporcional	43
Figura 6-22 Diagrama de conexiones del LM350 [37]	44
Figura 6-23 Circuito del Control de la Regulación de Voltaje	45

Lista de Tablas

Tabla 5-1 Exactitud para distintos valores de <code>n_neighbors</code>	25
Tabla 5-2 Reporte de Clasificación KNN	26
Tabla 5-3 Reporte de Clasificación Random Forest	26
Tabla 5-4 Exactitud para cada tipo de Naïve Bayes	26
Tabla 5-5 Reporte de Clasificación Naïve Bayes	27
Tabla 5-6 Exactitud para distintos tipos de kernel de SVM	27
Tabla 5-7 Reporte de Clasificación SVM	27
Tabla 5-8 Reporte de Clasificación MaxVoting	28
Tabla 6-1 Comparación entre el reporte de clasificación de los distintos algoritmos	28

Glosario

- **EMG:** Electromiografía.
- **AUC:** Área Bajo la Curva
- **ROC:** Característica Operativa del Receptor
- **ODS:** objetivos de desarrollo sostenible
- **OMT:** Organización Mundial del Trabajo
- **KNN:** K Nearest neighbors
- **SVM:** Support vector Machine
- **IoMT:** internet of Medic Things
- **ANN:** Artificial Neural Network
- **CNN:** Convolutional Neural Network
- **IDE:** Entorno de desarrollo integrado
- **Outlier:** Valor atípico

1. Introducción

La integración de tecnologías avanzadas como el *machine learning* en el diseño y desarrollo de dispositivos biomédicos ha emergido como un campo de investigación crucial y prometedor en la intersección entre la ingeniería, la medicina y la informática. En este contexto, el diseño de exoesqueletos controlados por señales de electromiografía (EMG) representa un área de estudio particularmente fascinante y de gran relevancia, con el potencial de transformar radicalmente la calidad de vida de individuos con discapacidades motoras, rehabilitación y prevención de heridas.

Los exoesqueletos son dispositivos externos que mejoran la fuerza, resistencia y coordinación del movimiento humano, han capturado la imaginación de científicos, ingenieros y profesionales de la salud debido a su capacidad para amplificar las capacidades humanas y restaurar la movilidad en aquellos que enfrentan limitaciones. Sin embargo, el desafío fundamental radica en el desarrollo de sistemas de control eficientes y precisos que permitan una interacción intuitiva y natural entre el usuario y el exoesqueleto.

En este contexto, las señales de electromiografía, que son señales eléctricas producidas por la activación muscular, han surgido como una vía prometedora para la interfaz hombre-máquina en el control de exoesqueletos. La capacidad de capturar la actividad eléctrica de los músculos en tiempo real ofrece una ventana hacia la intención de movimiento del individuo, lo que permite una interacción más fluida y adaptable con el dispositivo.

Sin embargo, la complejidad inherente de las señales EMG, junto con la variabilidad inter e intraindividual, plantea desafíos significativos en términos de procesamiento y decodificación de estas señales en comandos de control precisos para el exoesqueleto. Afortunadamente, existen datasets los cuales albergan gran cantidad de información la cual puede ser vital para el procesamiento y control a partir de estas señales. Junto al *machine learning*, se puede lograr analizar esta gran cantidad de información para conseguir un control preciso y adecuado para gran cantidad de aplicaciones.

En esta tesis de grado, se propone explorar y desarrollar métodos de implementación de técnicas de *machine learning* en el diseño y control de un mecanismo exoesqueleto basado en señales EMG para la prevención de lesiones músculo-esqueléticas en el campo industrial, aplicando un enfoque interdisciplinario que integra principios de ingeniería biomédica y ciencias de la computación, nuestro objetivo es avanzar en la comprensión y aplicación de estos métodos para mejorar la funcionalidad y eficacia de los exoesqueletos como herramientas de asistencia y rehabilitación.

Se aspira a contribuir al conocimiento académico en el campo emergente de las señales EMG y la ingeniería electrónica e impulsar el desarrollo de tecnologías innovadoras que tengan un impacto significativo en la vida de las personas.

1.1. Planteamiento del Problema

Las lesiones y enfermedades musculares en el campo industrial aún son demasiado comunes a pesar de que muchas de las actividades repetitivas y arduas ya han sido remplazadas por la automatización. Según la Organización Mundial del Trabajo (OMT), el 59 % de las patologías producto de actividades profesionales tienen que ver con lesiones musculoesqueléticas y estas lesiones en el caso de Colombia, representan el 65 % [1]. Un ejemplo nacional de la presencia de estas lesiones se evidencia en el estudio [2] realizado en una empresa de industria farmacéutica colombiana, que evidenció la prevalencia de síntomas relacionados a enfermedades esqueleto musculares en el 79,2 % de sus trabajadores. Los trabajos pesados como la minería, industria y construcción son más propensos a causar problemas en el sistema musculoesquelético. Según el Ministerio de Trabajo, el promedio de accidentes laborales es del 7 %, pero los trabajadores del campo de construcción tienen un promedio mayor del 10,5 % [3] por lo que esta profesión es una de las que puede generar mayor cantidad de lesiones laborales.

Otro grupo de trabajadores quienes son propensos a sufrir lesiones corresponden a la industria manufacturera, históricamente este es de los sectores con mayor cantidad de lesionados, siendo las lesiones musculoesqueléticas las más prominentes. En un país desarrollado como es Estados Unidos, se presentaron 373 mil incidentes de lesiones en la industria manufacturera en el año 2020; esto representa que de cada 100 trabajadores 3 sufrieron lesiones [4]. La relación principal que está presente en los trabajos pesados es que los trabajadores se ven expuestos a actividades en las que toman posturas incómodas, levantan objetos pesados y realizan movimientos repetitivos. Todas estas acciones generan desgaste muscular e inevitablemente causarán alguna enfermedad o lesión muscular si se ejecutan excesivamente. Una de las enfermedades musculares más comunes es la tendinitis la cual es la inflamación severa de un tendón, esta puede ser causada principalmente por la repetición de un mismo movimiento durante un tiempo extendido, lo que causa tensión excesiva y en algunos casos desgarro del tendón. Esta enfermedad es frecuente en personas cuyos empleos implican repetitividad, sostener posiciones incómodas, mantener los brazos por encima de la cabeza por mucho tiempo o estar expuesto a vibraciones y movimientos forzados. Los síntomas son un dolor sordo que se produce específicamente en donde el tendón afectado se adhiere al hueso, especialmente cuando se mueve la extremidad lesionada y sensibilidad e hinchazón en la zona [5].

Otro tipo de lesión muy común son los desgarros musculares, los cuales suponen alrededor del 30 % de las lesiones en la población general. Esta consiste en la rotura de un músculo o tendón causado por el exceso de esfuerzo en el tejido muscular, que generalmente ocurre al levantar objetos pesados. Los desgarros musculares pueden tener tres grados de severidad: El desgarro muscular de primer grado, se caracteriza por daños ligeros en fibras musculares y cuya recuperación no es tardía. El desgarro muscular de segundo grado se refiere a rotura parcial de fibras musculares y su recuperación puede durar unas cuantas semanas.

Finalmente, está el desgarro muscular de tercer grado, que es una rotura total del músculo y necesita tratamientos de atención profunda. Los síntomas varían dependiendo del grado del desgarro por lo que van desde un simple dolor y sensibilidad en la zona, hasta un dolor fuerte y agudo acompañado por inflamación y la pérdida de funcionalidad del músculo afectado [6].

De acuerdo con los aspectos mencionados anteriormente, se ve la necesidad de crear soluciones que puedan soportar la problemática que enfrentan las personas que se desempeñan en actividades que requieren grandes cargas musculares. Una alternativa se encuentra en el análisis de las señales de electromiografía (EMG) y las técnicas de *machine learning*, las cuales permiten hacer predicciones sobre esfuerzos musculares peligrosos y así poder tomar medidas preventivas para evitar lesiones. De esta manera surge la siguiente pregunta problema:

¿Cómo implementar técnicas de *machine learning* en el uso de señales de electromiografía (EMG) para el desarrollo del controlador de un exoesqueleto el cual reduzca la tensión muscular en los bíceps que enfrentan los trabajadores de industrias pesadas?

1.2. Objetivos

1.2.1. Objetivo General

Proponer una estrategia de control de un exoesqueleto que permita la reducción de la tensión muscular que soportan los bíceps, utilizando técnicas de clasificación de *machine learning* basadas en señales EMG

1.2.2. Objetivos Específicos

- Identificar los algoritmos de *machine learning* aplicables en el análisis y clasificación de posiciones de los bíceps a través de señales EMG.
- Crear modelos de *machine learning* para la clasificación de movimientos correctos e incorrectos de los bíceps utilizando un *dataset* existente relacionado con señales EMG.
- Distinguir las aplicaciones de los modelos creados para una estrategia de control de un exoesqueleto que permita la reducción de la tensión muscular en los bíceps.

1.3. Justificación

Las señales de electromiografía (EMG) sirven para detectar y examinar la señal eléctrica generada por los nervios y músculos en los estados de reposo y activación muscular. La posibilidad de medir la contracción y estimulación de estos tejidos genera un gran potencial para el desarrollo de numerosas aplicaciones para mejorar la calidad de vida humana.

Actualmente, las aplicaciones más comunes se encuentran dentro del campo médico para la detección de enfermedades de forma no invasiva y el desarrollo de control para exoesqueletos de rehabilitación y prótesis para amputados, los cuales han facilitado el proceso de tratamiento y diagnóstico de diferentes condiciones médicas, todo esto gracias a la implementación de *machine learning* para el análisis y clasificación de datos. Un ejemplo de la ayuda que es capaz de otorgar el *machine learning* es el desarrollo de un algoritmo por un grupo de investigadores, el cual fue capaz de diferenciar 7 enfermedades a partir de un *dataset* de síntomas presentados en 1000 personas [7]. El resultado fue un algoritmo *Support Vector Machine* (SVM) el cual pudo diferenciar las enfermedades con una exactitud del 84 %, lo cual no solo hace más fácil y rápido la detección de las enfermedades, sino que también es capaz de desarrollarlo con gran precisión. Por lo que se ve la posibilidad de ampliar este campo del *machine learning* para el uso de señales EMG e implementar dichas señales para utilidades más cotidianas, las cuales podrían tratar varias enfermedades mediante la prevención del desgaste muscular.

La propuesta de este trabajo es abordar el estudio de las enfermedades musculoesqueléticas asociadas a los bíceps, que son una de las más representativas dentro del 59 % de las patologías diagnosticadas producto de actividades profesionales [1]. A este tipo de enfermedades son propensos los trabajadores de industrias pesadas como lo son la manufacturera y la de construcción, debido a que deben desarrollar levantamiento de objetos pesados y movimientos repetitivos, los cuales pueden causar desgarros musculares y tendinitis respectivamente. Con el ánimo de proporcionar herramientas tecnológicas que puedan mitigar los problemas anteriormente mencionados, se propone diseñar un control para un mecanismo exoesqueleto mediante el procesamiento de señales EMG a través de *machine learning* para así reconocer el esfuerzo requerido por la extremidad (brazo), con el fin de otorgar una fuerza relativa en esta, y así reducir el estrés en los músculos y tendones.

También se busca decrementar la fatiga producto del trabajo, ya que la energía requerida para realizar el movimiento se disminuiría. Este proyecto explora la forma en que el análisis de las señales EMG se puede aplicar en nuevas áreas y también representa una forma no invasiva a la hora de desarrollar las mediciones debido a que se hace a través de un método superficial, el cual también es fácil de instalar y remover.

1.4. Impacto Social

Este proyecto le apunta al Objetivo de Desarrollo Sostenible (ODS) número 3 de salud y bienestar [8], ya que se busca beneficiar a las personas expuestas a cargas físicas considerables en su actividad profesional, mediante un sistema que puede llegar a ser implementado para diagnosticar problemas relacionados con los riesgos laborales.

Este tipo de soluciones pretende mejorar la calidad de vida de personas con vulnerabilidad causada por sus actividades laborales y también busca la mejora de los espacios en los que los trabajadores se desempeñan. En Bogotá hay 188,6 mil trabajadores en el campo de la construcción y en Colombia hay en total 686,4 mil trabajadores [9], de los cuales muchos de estos se podrían ver beneficiados con el desarrollo de este proyecto.

También se busca beneficiar el Grupo de Estudio y Desarrollo en robótica (GED) de la Universidad Santo Tomás, ya que se plantea un nuevo tipo de proyecto en el que este grupo se puede enfocar, donde se abarcan las líneas de inteligencia computacional, la robótica, y los algoritmos de *machine learning* para el análisis de señales biomédicas

2. Estado del Arte

Para la demostración del potencial que tiene el uso de señales EMG en diferentes aplicaciones, y para la contextualización, explicación del funcionamiento, obtención y análisis de estas señales, se realizaron consultas de investigaciones previas en bases de datos especializadas cuyos artículos abarcan diferentes formas de usar las señales de electromiografía.

2.1. Procesamiento y análisis de las señales EMG

La adecuación y procesamiento de la señal recibida es un paso importante para así eliminar ruido y obtener una respuesta lo más exacta posible para el mecanismo propuesto, para ello se investigó acerca de formas de como otros investigadores mejoraron el proceso de medición y clasificación de estas señales. Una de las aplicaciones más usadas es el reconocimiento de gestos mediante medición de pulsos musculares con sensores EMG.

Un enfoque interesante fue dado por un grupo de investigadores quienes sugieren la implementación de blockchain en el internet de las cosas médicas (IoMT) para el análisis y reconocimiento de patrones para reconocimiento de gestos [10]. Ellos usaron un filtro de peine mejorado y un modelo de mezcla para reducir ruido de señales EMG superficiales. Lograron demostrar que su método propuesto redujo efectivamente reconocimientos falsos y alcanzó un reconocimiento de gestos dinámico-continuo de alta precisión.

Otro grupo de investigadores se propuso usar dichos reconocimientos para detectar las posiciones que puede adquirir la mano midiendo la activación de los músculos del antebrazo, para ello usaron transformación suave de paquetes wavelet y un algoritmo de red neuronal convolucional híbrido (CNN) [11]. Con esta transformada lograron descomponer la señal EMG en el dominio tiempo-frecuencia obteniendo resultados favorables ya que la señal se pudo procesar más rápido que otros métodos de descomposición. Además, el algoritmo CNN pudo obtener una exactitud promedio del 92,159 % lo cual superó otros estudios de investigación similares al evaluar los resultados en al *dataset* público NapiroDB5.

Se han buscado otras formas de clasificar las señales obtenidas para reconocimientos más precisos, uno de esos métodos fue implementado a partir de redes neuronales artificiales (ANN) apoyado por un algoritmo SVM [12]. Se desarrolló un experimento con 10 participantes saludables donde se consiguió que la ANN alcanzara la media de precisión más alta y la varianza de medidas entre participantes más baja a comparación de otros estudios similares. Otro enfoque que se le dio fue el análisis no lineal de señales superficiales EMG de los bíceps mediante la separación entre la parte caótica y no caótica de la señal EMG usando dimensión fractal (FD) usando el método petrosiano y dimensión de correlación (CD) [13]. Los resultados demostraron que en las aplicaciones en las que se usaron CD y FD se alcanzó una exactitud en la separación de la parte no caótica de la señal

con una exactitud del 98,97 % por lo que por primera vez se logró excluir la parte caótica de la señal EMG de los bíceps.

Con esta sección de la investigación se logró identificar que es de gran importancia la limpieza y adecuación de la señal para una aplicación de exoesqueleto ya que se requiere de gran exactitud en la toma de datos, para ello se deberá aplicar filtros para reducir el ruido para minimizar los reconocimientos falsos y lograr una respuesta adecuada del sistema.

2.2. Robótica aplicada

Con la habilidad de detectar activación muscular, surge la posibilidad de usar estas señales para lograr guiar mecanismos robóticos, prostéticos y exoesqueletos para así lograr un movimiento lo más natural posible.

Una de estas posibilidades fue abordada en 2019 en el desarrollo de una replicación a tiempo real de movimientos del brazo a partir de señales EMG superficiales [14]. Para ello se propusieron nueve movimientos de esta extremidad y se decidió controlar las articulaciones de la muñeca y el codo del brazo robótico con respecto a las registradas en el brazo real. Las señales se tomaron como entrada para un algoritmo SVM aplicado con una función de base radial se logró una exactitud en la imitación del brazo real del 93,3 %.

Otro grupo logró una hazaña similar mediante el mejoramiento del control de una mano robótica con el uso de un controlador Fuzzy-PI también usando SVM para la clasificación de las señales [15], los resultados mostraron un movimiento notablemente idéntico a una mano humana. En el ámbito de exoesqueletos, se encontró un estudio sobre un control usando sensores EMG para un exoesqueleto para las extremidades superiores, más específicamente los hombros [16]. Estas señales se pudieron clasificar para lograr reconocer gestos para beber, abrir una puerta, abducción del músculo y posición de descanso usando *machine learning*. Estas señales fueron aplicadas a un exoesqueleto con una estructura simple y ligera el cual tiene motores en zonas de articulaciones (codo, hombro y espalda superior). Se observó que los algoritmos de más precisos fueron ANN y SVM, los cuales obtuvieron una exactitud del $96 \% \pm 3, 8 \%$ en configuración offline, el estudio demostró que el *machine learning* provee un acercamiento confiable para reconocimiento de patrones de movimiento del hombro y para un control a tiempo real de un exoesqueleto.

Otro exoesqueleto desarrollado fue el NESM [17], el cual posee 4 articulaciones rotacionales activas (3 en el hombro y una en el codo) usando actuadores elásticos en serie y de alto torque. Este exoesqueleto es usado para la rehabilitación de pacientes que han sufrido un derrame cerebral.

Para prótesis, es importante un procesamiento preciso para asegurar que esta sea lo más parecido posible a un brazo real. Se ha abarcado este problema en un estudio el cual se desarrolló un mejoramiento vía técnicas de optimización de filtrado con la intención de crear un control de prótesis intuitivo por reconocimiento de

patrones para amputados [18]. La importancia de un filtrado mejorado deriva a que las señales EMG tomadas en zonas amputadas es limitada y usualmente se adquiere una gran cantidad de ruido lo que es un reto decodificar el movimiento de brazo intencionado por el paciente. Se descubrió que el filtro Weiner (WF) alcanzó un incremento de hasta el 18,64 % frente a otros tipos de filtro por lo que se logró mejorar sustancialmente la clasificación de señales EMG que determinaban las intenciones de movimiento en extremidades amputadas. Con la perspectiva útil de trabajos en campos de la robótica, se encontró la gran importancia que tiene los algoritmos de *machine learning* para proyectos de este estilo para así desarrollar el reconocimiento de movimientos.

Se pudo concluir que los algoritmos más usados por su exactitud y desempeño fueron SVM y ANN, por lo que estos van a ser un foco principal a la hora del desarrollo del proyecto.

2.3. Estudio de Enfermedades

Las señales EMG también pueden ser usadas en un ámbito médico, estas señales son usadas regularmente por especialistas para analizar irregularidades musculares.

En la investigación [19], se planteó el uso de señales EMG para la clasificación de miopatía, esclerosis amiotrófica lateral y señales saludables. Usaron un método de modo de descomposición empírica (EMD) para descomponer las señales EMG en funciones de modo intrínsecas y usarlas para hacer la clasificación de las enfermedades. Después se entrenó y validó un algoritmo con una red neuronal de avance (FFNN), una máquina de soporte vectorial (SVM) y un árbol de decisión. el resultado fue una clasificación de exactitud del 99,53 %, sensibilidad del 99,25 % y especificidad del 99,60 %.

La investigación realizada en [20], se enfocó en el análisis y evaluación de la sarcopenia, en donde se tomaron medidas en 32 pacientes de las señales EMG en los músculos gastrocnemio lateral y tibial anterior para determinar 3 niveles de confianza los cuales determinan la gravedad de la enfermedad, con lo que se consiguió alto nivel de precisión de clasificación de dichos niveles de confianza usando el clasificador de SVM.

También se ha logrado el uso de estas señales para el desarrollo de un método no intrusivo para diagnosticar la estenosis traqueal, la cual es una enfermedad que genera el estrechamiento de la tráquea lo que dificulta la respiración [21]. Esto se logró mediante la clasificación de las señales EMG usando un algoritmo clasificador aprendizaje adaptativo sensible a costos (AdaCSL) el cual al ser implementado en dos experimentos con pacientes logró clasificar dicha enfermedad respiratoria con una mayor precisión que otros clasificadores.

En esta sección se pudo resaltar la importancia de los algoritmos de clasificación mediante *machine learning* para la clasificación de anomalías en la activación muscular y la posibilidad de usar formas no intrusivas para medir dichas activaciones.

3. Marco Teórico

3.1. Señales de Electromiografía

Las señales de electromiografía (EMG) son aquellas que representan la actividad eléctrica producida por los músculos. Se generan cuando las células musculares, llamadas fibras musculares, se despolarizan, lo que significa que hay un cambio en la distribución de cargas eléctricas a través de la membrana celular.

La actividad eléctrica de las fibras musculares se transmite a la superficie de la piel, donde puede ser detectada por electrodos. Las señales EMG tienen una amplitud típica de hasta 6mV y tienen una frecuencia útil que oscila entre 20 y 500 Hz. La amplitud de la señal es proporcional a la fuerza de contracción muscular: Cuanto mayor sea esta fuerza de contracción, mayor será la amplitud de la señal medida [22].

Estas señales pueden ser utilizadas para el diagnóstico de numerosas enfermedades neuromusculares como la distrofia y la miopatía. También se usan para evaluar la respuesta muscular frente a numerosos estímulos que pueden ser usados para procesos de rehabilitación. Finalmente, una de las utilidades más interesantes de las señales EMG es en la que se enfoca este trabajo: el control de prótesis y ortesis

3.2. Algoritmos:

Un algoritmo en *machine learning* representa un enfoque formal y estructurado para abordar problemas complejos mediante la utilización de datos. Estos procedimientos están meticulosamente diseñados para permitir que un sistema informático genere predicciones, identifique patrones y tome decisiones autónomas basadas en la información disponible. En lugar de seguir instrucciones específicas para cada tarea, estas metodologías permiten que las máquinas adquieran conocimiento y mejoren su rendimiento a través de la experiencia y la exposición a conjuntos de datos relevantes. En esencia, los algoritmos de *machine learning* actúan como herramientas poderosas que capacitan a los sistemas informáticos para aprender de manera eficiente y automatizada, abriendo nuevas posibilidades para la resolución de problemas en una variedad de campos, desde la medicina hasta las finanzas y más allá [23].

Estos algoritmos pueden ser clasificados principalmente en 3 tipos:

3.2.1. Algoritmos de Aprendizaje por Refuerzo:

Este enfoque se basa a las interacciones de un agente con un entorno en el cual aprende mediante la toma de acciones y recibe una retroalimentación en forma de recompensas o castigos que indican al algoritmo que tan buena fue esa acción desarrollada. Se basa en que el agente aprende a partir de sus experiencias e interacciones continuas con su entorno al usar técnicas como la exploración y explotación para mejorar su desempeño [24].

Los componentes principales de este tipo de algoritmos son:

- **Agente:** El agente es el aprendiz o el algoritmo de aprendizaje por refuerzo. Su objetivo es aprender a tomar decisiones óptimas para maximizar una recompensa acumulada a lo largo del tiempo.
- **Entorno:** El entorno es el mundo en el que opera el agente. Proporciona el contexto en el que el agente toma decisiones y recibe retroalimentación.
- **Acciones:** El agente puede tomar una serie de acciones en cada paso de tiempo. Estas acciones pueden afectar al estado del entorno y, por lo tanto, influir en las recompensas que el agente recibe.
- **Estado:** El estado del entorno representa la situación actual en la que se encuentra el agente. Es una representación de la información relevante del entorno en un momento dado.
- **Recompensas:** Las recompensas son señales de retroalimentación que el agente recibe del entorno después de tomar una acción en un estado particular. Indican qué tan buena o mala fue la acción tomada por el agente.

El aprendizaje por refuerzo se lleva a cabo a través de ciclos repetidos de interacción entre el agente y el entorno. El proceso general inicia con el agente observando el estado actual de su entorno para después seleccionar una acción a partir de una política de acciones establecidas. Una vez realizada la acción, el entorno cambia de estados en respuesta de dicha acción y el agente actualiza su política en función de la experiencia acumulada respecto a si su experiencia fue positiva o negativa.

El objetivo del agente es desarrollar una política que le permita tomar las mejores acciones en cada estado, con el fin de maximizar la recompensa acumulada a lo largo del tiempo. Esto se logra mediante la aplicación de técnicas de aprendizaje, tales como los métodos basados en valores (por ejemplo, Q-Learning) o basados en políticas (como la Programación Dinámica o los Métodos de Monte Carlo), que permiten al agente mejorar continuamente su toma de decisiones a medida que interactúa con el entorno. A continuación, se explica el método de aprendizaje por refuerzo más popular:

3.2.1.1. Q-learning

es un enfoque de aprendizaje automático que permite a un modelo aprender y mejorar iterativamente con el tiempo mediante la toma de acciones correctas. Es una forma de aprendizaje por refuerzo, donde el modelo se entrena para imitar la manera en que los animales o los niños aprenden: las buenas acciones se recompensan y refuerzan, mientras que las malas acciones se desaconsejan y penalizan.

Q-learning utiliza un enfoque *off-policy*, lo que significa que el agente puede aprender una política óptima independientemente de la política seguida durante el aprendizaje. Esto se logra utilizando *Q-values*, que son los valores futuros esperados para cada acción y se almacenan en una tabla *Q-table*.

En Q-learning, el proceso de aprendizaje sigue el ciclo de estado-acción-recompensa-estado-acción. El agente toma una acción en un estado particular, recibe una recompensa, observa el nuevo estado resultante y actualiza los *Q-values* en la tabla *Q-table* en función de la recompensa recibida y el valor máximo de *Q* del nuevo estado. La actualización de los *Q-values* se desarrolla mediante la ecuación de Bellman:

$$\hat{Q}(s, a) = Q(s, a) + \alpha [R + (\lambda * \max_{a'} Q(s', a')) - Q(s, a)]$$

Donde:

$Q(s, a) \rightarrow$ Valor actual

$\alpha \rightarrow$ tasa de aprendizaje

$R \rightarrow$ Recompensa

$\lambda \rightarrow$ Tasa de descuento

$\max Q(s', a') \rightarrow$ Valor óptimo esperado

3.2.2. Algoritmos no Supervisados:

Los algoritmos no supervisados son una clase de técnicas de aprendizaje automático que se utilizan para analizar y modelar datos sin la necesidad de etiquetas o supervisión explícita. El objetivo principal de estos algoritmos es descubrir patrones, estructuras o relaciones ocultas en los datos. Estos algoritmos suelen ser usados para varias aplicaciones como la segmentación o agrupamiento (*clustering*) de clases o reducción de dimensionalidad para facilitar visualización y análisis de datos complejos [25].

3.2.2.1. Clustering

El clustering es una técnica de aprendizaje automático no supervisado que agrupa datos en subconjuntos llamados clústeres, de tal manera que los elementos dentro de un mismo clúster son más similares entre sí que a los elementos de otros clústeres. La similitud se mide según una métrica o criterio específico, que puede variar dependiendo del algoritmo de clustering utilizado y del tipo de datos. A continuación, se explica brevemente cómo funciona las métricas o criterios mas comunes para desarrollar la agrupación de los datos a partir de la similitud entre ellos:

- **K-means:** Divide los datos en K grupos basándose en la proximidad de los puntos de datos al centroide de cada grupo.

- **Hierarchical Clustering (Clustering Jerárquico):** Crea una jerarquía de clústeres, donde cada punto de datos comienza como un clúster individual y se fusiona con otros clústeres a medida que se sube en la jerarquía.
- **DBSCAN:** Identifica clústeres basados en la densidad de puntos en el espacio de características, útil para datos con formas arbitrarias.

3.2.2.2. Reducción de Dimensionalidad

La reducción de dimensionalidad es una técnica que busca disminuir el número de variables aleatorias bajo consideración, obteniendo un conjunto de variables principales más manejable. Esto se realiza sin perder demasiada información importante del conjunto de datos original. La reducción de dimensionalidad es crucial para manejar datos de alta dimensionalidad, mejorar la visualización, reducir el ruido, y aumentar la eficiencia computacional. Los principales algoritmos de reducción de dimensiones son:

- **Principal Component Analysis (PCA):** Se encarga de encontrar una nueva base ortogonal en la que las primeras pocas dimensiones retengan la mayor varianza de los datos originales. Esto lo logra centrando los datos alrededor de un origen. Después, calcula la matriz de covarianza de los datos centrados y calcula los eigenvalores y eigenvectores de dicha matriz. Procede seleccionando los eigenvectores correspondientes a los mayores eigenvalores para así poder proyectar los datos originales en el espacio de menor dimensión usando estos eigenvectores.
- **t-Distributed Stochastic Neighbor Embedding (t-SNE):** se encarga de mapear datos de alta dimensionalidad a un espacio de baja dimensionalidad (generalmente 2D o 3D) preservando las relaciones de proximidad locales entre puntos. Primero, calcula las probabilidades de proximidad entre puntos en el espacio de alta dimensión. Después, Define un mapeo en el espacio de baja dimensión y calcular las probabilidades correspondientes. Finalmente, minimiza la divergencia entre las distribuciones de probabilidad en los dos espacios.
- **Autoencoders:** Son redes neuronales usadas para aprender una representación comprimida de los datos de entrada. Esto se refiere, a que el algoritmo entrena una red neuronal con una capa que tenga menos neuronas que la capa de entrada y salida con el fin de reproducir la entrada en la salida y así forzar a la red neuronal a aprender una representación reducida de los datos, reduciendo así la dimensionalidad.

3.2.3. Algoritmos Supervisados:

es un enfoque en donde el modelo se entrena a partir de datos previamente etiquetados con la clasificación deseada. El modelo toma unos datos para

entrenamiento con el fin de entender cómo afectan las características a la clasificación de cada etiqueta y así poder entender la relación entre estos dos. Una vez el modelo sea entrenado, este puede predecir las clases a partir de nuevos datos de entrada basándose en las relaciones aprendidas previamente en la etapa de entrenamiento.

Para el desarrollo de este proyecto, se ha optado por la implementación de algoritmos supervisados. Estos algoritmos se destacan por su capacidad para ofrecer una alta precisión al trabajar con *datasets* que están adecuadamente etiquetados. Además, suelen ser más fáciles de comprender e implementar en comparación con otros enfoques. Esto se debe a la amplia disponibilidad de opciones de algoritmos que pueden ser utilizados para abordar el problema específico en cuestión.

Una ventaja adicional de utilizar algoritmos supervisados es que los *datasets* disponibles para este proyecto ya contienen etiquetas de clasificación, lo que los hace adecuados para ser utilizados con este tipo de algoritmos. Esta característica simplifica considerablemente el proceso de implementación y entrenamiento del modelo, permitiendo un avance más rápido en el desarrollo del proyecto.

Los siguientes son los algoritmos supervisados que se implementarán para el desarrollo de la clasificación de movimientos correctos e incorrectos de los bíceps:

3.2.3.1. Algoritmo KNN

El algoritmo *K-Nearest Neighbors* (KNN) es un método de aprendizaje supervisado utilizado para la clasificación. Durante el entrenamiento, este algoritmo almacena los datos junto con sus respectivas etiquetas de clase. Luego, calcula la distancia entre un punto específico y todos los demás puntos de datos en el conjunto. Posteriormente, identifica los K puntos más cercanos basándose en estas distancias, los cuales son denominados vecinos. Finalmente, el algoritmo asigna la etiqueta de clase más común entre los K vecinos más cercanos al punto seleccionado. Una representación del funcionamiento de este algoritmo puede ser observada en la Figura 3-1.

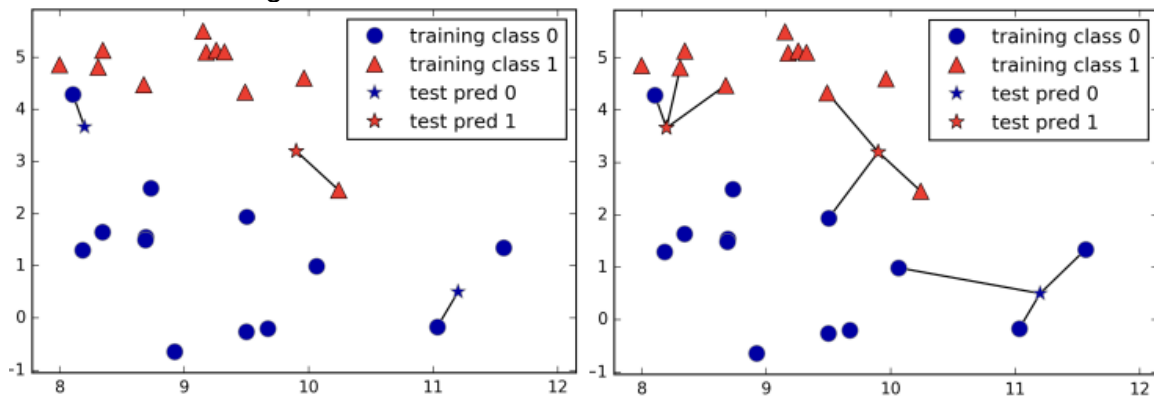


Figura 3-1 Funcionamiento algoritmo KNN [26]

En la Figura 3-2, se observa que cuando se utiliza un solo vecino, se logra una mayor precisión en la clasificación de los datos. Sin embargo, esto conlleva a una mayor complejidad, como se puede observar en la frontera de decisión generada. Por otro lado, al aumentar la cantidad de vecinos, se obtiene una frontera de decisión más suave, lo que resulta en un modelo más simple. No obstante, este enfoque suele implicar un sacrificio en la precisión de la clasificación.

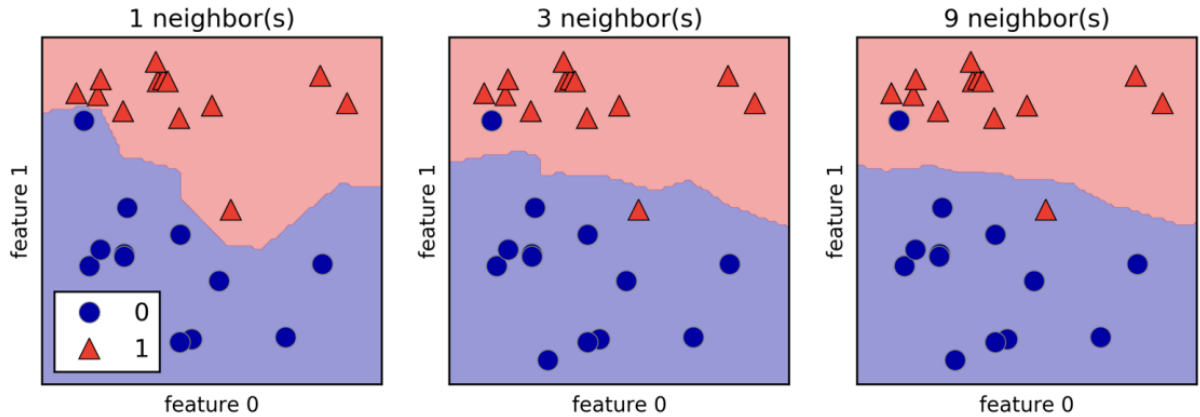


Figura 3-2 Diagrama de frontera de decisión KNN [26]

Para calcular la distancia entre el punto de consulta a los vecinos se usan tres métodos:

- **Distancia Euclidiana:** Es la distancia cartesiana entre los dos puntos, puede ser visualizada como la distancia de una línea recta que une los puntos considerados.

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (1)$$

- **Distancia Manhattan:** En vez de calcular la distancia en una línea recta entre los puntos, se calcula la suma de las diferencias absolutas de las coordenadas cartesianas.

$$d(x, y) = \sum_{i=1}^n |x_i - y_i| \quad (2)$$

- **Distancia Minkowski:** Esta fórmula implementa una variable p la cual dependiendo de su valor se calcula la distancia Manhattan o la distancia Euclidiana.

$$d(x, y) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}} \quad (3)$$

Se seleccionó este algoritmo para implementarlo en el proyecto debido a su simplicidad y facilidad de comprensión, ya que su lógica se centra en encontrar los

puntos de datos más cercanos en el espacio. Además, su naturaleza paramétrica lo convierte en una opción robusta y adaptable, ya que no hace suposiciones explícitas sobre la distribución de los datos subyacentes. Esta característica es especialmente útil en *datasets* que pueden presentar ruido y valores atípicos, como el *dataset* seleccionado para este proyecto.

3.2.3.2. Algoritmo Random Forest

En *machine learning*, los árboles de decisión son modelos que operan dividiendo de manera repetida el conjunto de datos en subconjuntos más pequeños, basándose en características específicas. Su objetivo final es construir una estructura de árbol que pueda predecir la variable objetivo con precisión. A partir de este concepto se crea el algoritmo Random Forest.

El algoritmo Random Forest es una técnica que combina múltiples árboles de decisión individuales para mejorar la precisión y la capacidad de generalización del modelo. Este algoritmo consta de numerosos árboles de decisión, cada uno de los cuales es único debido a la introducción de aleatoriedad durante su construcción. En cada nodo, se elige la mejor división utilizando alguna medida de impureza, hasta alcanzar la profundidad máxima especificada. Para clasificar los datos de entrada, cada árbol genera una votación, y la clase más frecuente entre todos los árboles se considera la predicción final [27].

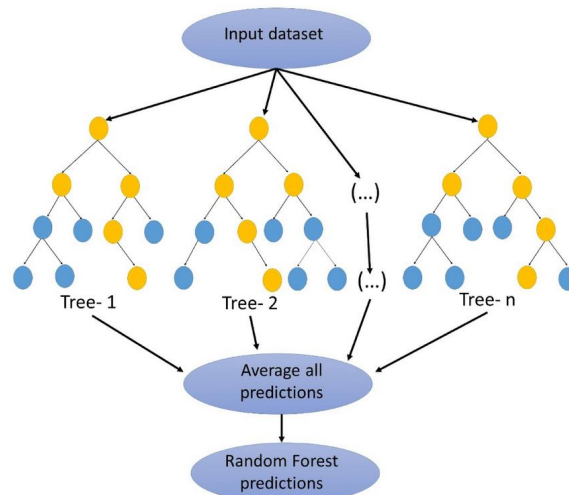


Figura 3-3 Diagrama de Árboles Random Forest [27]

A diferencia de los árboles de decisión individuales, Random Forest tiende a mitigar el sobreajuste (*overfitting*) gracias a la implementación de un gran número de árboles de decisión. Esta estrategia contribuye a reducir la varianza general y el error de predicción del modelo. Sin embargo, una desventaja de este algoritmo radica en que, debido a la inclusión de numerosos árboles de decisión, se obtiene una predicción más compleja y el proceso de predicción puede volverse más lento, especialmente con *datasets* de gran tamaño, dado que la cantidad de procesamiento requerida aumenta.

La robustez de Random Forest frente al sobreajuste y los valores atípicos, junto con su alta precisión, análisis profundo y eficiencia en *datasets* grandes, lo convierten en un candidato destacado para obtener una alta exactitud en la clasificación de la calidad del movimiento para el desarrollo de este proyecto.

3.2.3.3. Algoritmo Naïve Bayes

El algoritmo Naïve Bayes es un método supervisado utilizado en tareas de clasificación, cuyo funcionamiento se basa en el teorema de Bayes. Este teorema establece una relación entre la probabilidad condicional de un evento A dado otro evento B, en términos de la distribución de probabilidad del evento B dado el evento A y la distribución marginal solo del evento A [28][29]. Matemáticamente, se expresa de la siguiente manera:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (4)$$

Inicialmente, el algoritmo calcula la probabilidad de las clases y las probabilidades condicionales de las características para cada clase. Para ello, divide la frecuencia de cada clase entre el número total de muestras y la frecuencia de cada valor de la característica para una clase dada, entre la frecuencia total de dicha clase. Luego, utiliza el teorema de Bayes para calcular la probabilidad posterior de cada clase dada las características observadas. Esta probabilidad posterior se obtiene multiplicando las probabilidades condicionales y la probabilidad de las clases. La clase con la probabilidad posterior más alta se determina como la predicción final para el nuevo punto de datos.

Existen tres tipos principales de clasificadores Naïve Bayes:

- **Gaussiano:** Se asume que los valores continuos asociados a cada característica siguen una distribución Gaussiana.
- **Bernoulli:** Este tipo de clasificador se utiliza cuando las características son variables binarias que describen las entradas.
- **Multinomial:** Se emplea una distribución multinomial para generar eventos a partir de vectores de características que representan las frecuencias de ocurrencia.

Se espera que este algoritmo ofrezca un equilibrio entre una eficiencia computacional óptima y una precisión aceptable. Esto se debe a que el algoritmo Naïve Bayes es conocido por su simplicidad y facilidad de implementación, lo que lo convierte en una opción atractiva para aplicaciones con recursos computacionales limitados. Además, muestra un buen rendimiento en situaciones donde los datos están incompletos, aprovechando la probabilidad condicional de cada clase en su funcionamiento.

Sin embargo, es importante tener en cuenta que este algoritmo asume la independencia entre las características, una suposición que puede no ser realista

en casos reales. Esta limitación puede afectar su capacidad para generar predicciones altamente precisas en ciertas circunstancias.

3.2.3.4. Algoritmo SVM

El algoritmo SVM (*Support Vector Machine*) es una técnica versátil utilizada tanto para clasificación como para regresión en el ámbito del *machine learning*. Su objetivo principal es encontrar el hiperplano óptimo en un espacio n-dimensional que pueda separar eficientemente los datos en diferentes clases. Este hiperplano se selecciona de manera que maximice la separación entre las clases, lo que permite una clasificación más precisa.

Durante este proceso, el algoritmo SVM identifica los datos que están más cerca del hiperplano de decisión y los utiliza para determinar su ubicación y orientación. Estos datos son conocidos como vectores de soporte, y son fundamentales para definir el hiperplano de separación. Por otro lado, los datos que se encuentran más alejados de su clase con respecto al hiperplano seleccionado son ignorados por el algoritmo, ya que se consideran *outliers* o atípicos.

El objetivo del algoritmo SVM es encontrar un hiperplano que tenga la menor cantidad posible de *outliers*, lo que garantiza una clasificación más precisa y robusta. Al minimizar la influencia de estos puntos atípicos, el modelo puede generalizar mejor y adaptarse de manera más efectiva a nuevos datos, lo que resulta en un rendimiento óptimo en la clasificación de datos [30]. En la siguiente imagen se puede observar una representación del funcionamiento del algoritmo, en donde los ejes representan las características.

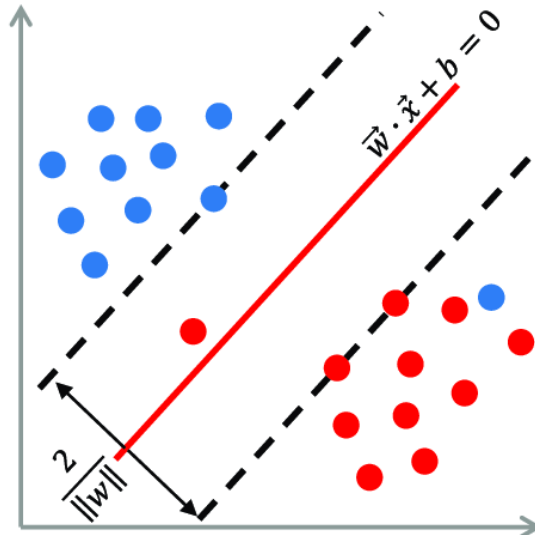


Figura 3-4 Funcionamiento Algoritmo SVM [31]

Hay dos enfoques para encontrar el hiperplano óptimo. El primero es mediante el uso de un algoritmo SVM lineal, que se emplea cuando los datos pueden ser separados por una línea recta o un hiperplano. La ecuación para calcular el hiperplano lineal es la siguiente:

$$K(\omega, b) = \omega^T x + b \quad (5)$$

Donde el parámetro ω representa el vector normal con respecto al hiperplano, y el parámetro b representa la distancia del hiperplano al punto de origen del vector normal ω .

El segundo enfoque es utilizando SVM no lineal, que se aplica cuando los datos no pueden ser separados por una línea recta en un espacio bidimensional. En este caso, el algoritmo puede emplear un kernel, que es una función matemática que se utiliza para transformar los datos de entrada en un espacio de características de mayor dimensión. Esta transformación permite que los datos sean más fácilmente separables por un hiperplano en el espacio de características de alta dimensión, incluso si no son linealmente separables en el espacio original de menor dimensión.

Existen varios tipos de kernels que se pueden utilizar. A continuación, se presenta la forma matemática en la que estos calculan el hiperplano óptimo:

$$\text{Polynomial: } K(\omega, b) = (\gamma \omega^T x + b)^N \quad (6)$$

$$\text{RBF: } K(\omega, b) = \exp(-\gamma |x_i - x_j|^n) \quad (7)$$

$$\text{Sigmoid: } K(\omega, b) = \tanh(\alpha x_i^T x_j + b) \quad (8)$$

Este algoritmo puede ofrecer una valiosa contribución al proyecto, ya que parece representar un equilibrio entre precisión y eficiencia computacional. SVM destaca por su capacidad de generalización de los datos, su habilidad para manejar datos no lineales a través del uso de kernels, y su capacidad para trazar fronteras de decisión complejas, características que son esenciales para el *dataset* usado en el proyecto. Además, es eficiente en memoria al utilizar solo un subconjunto de los puntos de datos en la función de decisión, por lo que reduce el consumo de recursos computacionales y garantiza una ejecución rápida.

3.2.3.5. Algoritmo MaxVoting:

Es una técnica de *ensemble learning* usada para problemas de clasificación de datos. Su fundamento radica en la premisa de que la combinación de múltiples algoritmos ofrece resultados superiores en comparación con su uso individual.

Inicialmente, se entrena cada modelo de clasificación de manera individual, ya sea utilizando diversos algoritmos o ajustando los parámetros del mismo algoritmo. Una vez entrenados, estos modelos se emplean para generar predicciones de manera independiente. Cada modelo produce una predicción de clase para cada instancia del conjunto de datos.

Posteriormente, se lleva a cabo el proceso de votación, en el cual se cuentan los votos correspondientes a la clase predicha por cada algoritmo. La clase que obtenga la mayoría de los votos se establece como la predicción final para la instancia en cuestión. De esta manera, se genera una lista de clases predichas, y se selecciona la que aparece con mayor frecuencia como la predicción definitiva.

En situaciones donde se produce un empate en la votación, se puede optar por realizar una selección aleatoria entre las clases empatadas o implementar métodos más complejos, como el voto ponderado. Esto asegura la resolución adecuada de casos donde no existe un consenso claro entre los modelos [32].

Esta técnica proporciona una mayor precisión y reduce el sobreajuste al combinar las predicciones de múltiples modelos. Además, es altamente versátil, ya que puede aplicarse a una amplia gama de modelos de clasificación, y es más estable debido a su menor sensibilidad a pequeñas variaciones en los datos. Por estas razones, se elige este algoritmo con el objetivo de obtener la máxima exactitud en las clasificaciones deseadas para este proyecto.

4. Diseño Metodológico

4.1. Fases y Metodología

Este proyecto tiene como objetivo el diseño de un controlador para un exoesqueleto con la implementación de *machine learning* a partir del análisis de señales EMG. En el presente trabajo se implementa un enfoque mixto, debido a que el enfoque cuantitativo del proyecto se evidencia en la recolección de datos de activación muscular a partir de *datasets*, y el enfoque cualitativo está presente en el análisis de estos datos y la interpretación que debe hacer el algoritmo para poder clasificarlos. Para un desarrollo ordenado y eficiente de la tesis se define en varias etapas la metodología a seguir:

4.1.1 Búsqueda Bibliográfica

Se lleva a cabo una búsqueda bibliográfica en la que se abarcarán los diferentes algoritmos de *machine learning* que puedan servir para la clasificación de posiciones correctas e incorrectas a través de señales EMG de los bíceps.

4.1.2. Búsqueda Datasets

En esta etapa se analizan diferentes *datasets* disponibles en repositorios públicos relacionados con señales EMG de los bíceps que puedan servir para clasificar distintos tipos de movimientos con respecto a si representan un riesgo para dicho músculo.

4.1.3. Desarrollo Algoritmo

Se procede a desarrollar en Python el algoritmo de *machine learning* a partir de los *datasets* disponibles y relevantes para el proyecto, y así generar las diferentes clasificaciones de movimientos. Posteriormente, se procederá a entrenar el algoritmo y a evaluar la exactitud de este.

4.1.4. Desarrollo Teórico Estrategia de Control del Exoesqueleto:

Este será un desarrollo teórico donde se dará la explicación de como funcionaría este controlador y su relación con el mecanismo exoesqueleto.

5. Desarrollo Conceptual

5.1. Selección del *Dataset*

El *dataset* se basa en el desarrollo establecido en la referencia [33], en donde los investigadores desarrollaron una toma de datos para identificar la calidad del movimiento de los bíceps.

Este *dataset* consiste principalmente en la señal EMG detectada por el sensor Myoware [36], diseñado para medir la activación muscular a través del potencial eléctrico de estos para ser usado en aplicaciones robóticas, prostéticas y otros sistemas de control. En este caso sirve para identificar si los bíceps están siendo sometidos a un esfuerzo excesivo debido a un estiramiento anormal del músculo causado por una técnica de levantamiento incorrecta.

No obstante, esta variable por sí sola no es completamente adecuada para determinar dicho sobre esfuerzo, ya que diferentes individuos pueden presentar niveles variados de activación muscular, influenciados por factores como la salud, la edad y la frecuencia de actividad física. Por esta razón, los investigadores implementaron un acelerómetro ubicado en la zona del deltoide medio, posicionado a un ángulo de 90°, con el fin de detectar movimientos que puedan indicar un levantamiento incorrecto. Este acelerómetro registra valores en los ejes X, Y y Z, los cuales son analizados para identificar patrones asociados a levantamientos inapropiados.

El conjunto de datos utilizado en este estudio se compone de 5000 muestras de la amplitud del sensor Myoware y de los 3 ejes del acelerómetro, obtenidas de 10 voluntarios que realizaron el ejercicio con mancuernas, con pesos que oscilaron entre 2 y 15 kilogramos. Estos voluntarios representan una variedad de perfiles, incluyendo individuos delgados con escasa experiencia en ejercicios de fuerza, personas con una condición física promedio que aspiran a asistir al gimnasio con regularidad, y personas con una constitución muscular más desarrollada que frecuentan el gimnasio de manera habitual.

Los datos recopilados se usaron para así obtener solo 2 clases, las cuales representan si el movimiento desarrollado fue correcto o incorrecto.

La recolección de datos se llevó a cabo mediante un Arduino Nano, y posteriormente se organizó la información en un archivo CSV, el cual se emplea en el presente proyecto como fuente de datos primaria.

Este *dataset* fue seleccionado por su similitud con el proyecto en cuestión, lo que simplifica considerablemente su análisis y procesamiento. En contraste, otros *datasets* encontrados presentaban variaciones en los músculos abordados y en variables innecesarias. Algunos carecían de clasificación, mientras que otros solo

proporcionaban la señal EMG, lo que implicaba la necesidad de un extenso preprocesamiento para adaptarlos a los requisitos de este trabajo.

5.2. Algoritmos Clasificatorios

Tras investigar los algoritmos disponibles, se avanza con el desarrollo de una prueba en Python para evaluar la exactitud de cada uno. En este proceso, se emplea la curva de Característica Operativa del Receptor (ROC) como métrica para evaluar el desempeño de estos modelos y realizar una comparación entre ellos. La Figura 5-1 es la gráfica obtenida:

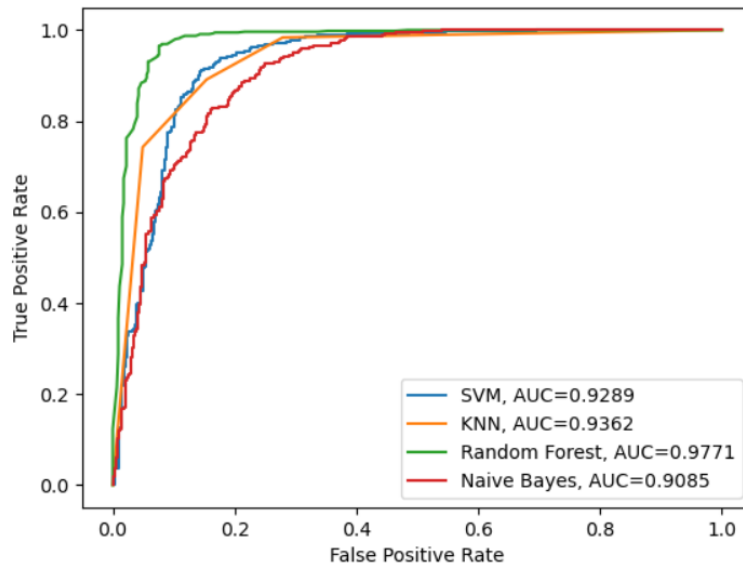


Figura 5-1 Gráfica ROC de los algoritmos

Con base en la Figura 5-1 y los valores del área bajo la curva (AUC), se puede determinar el orden de rendimiento de los algoritmos de la siguiente manera: El algoritmo con mejor desempeño, de acuerdo con la gráfica de ROC, es el Random Forest, con un AUC de 0,9771. En segundo lugar, se sitúa el algoritmo KNN, cuyo AUC es de 0,9362. A continuación, se encuentra el SVM con kernel rbf, con un AUC de 0,9289. En último lugar, se posiciona el algoritmo Naïve Bayes, con un AUC de 0,9085.

Estos resultados proporcionan una perspectiva sobre qué algoritmo podría funcionar mejor para la implementación en cuestión.

Tras realizar las pruebas pertinentes, se procede a implementar los diferentes algoritmos clasificatorios estudiados y disponibles en la librería SciKitLearn de Python. Se evalúa la precisión de cada uno de estos algoritmos para determinar cuál es el más adecuado para el proyecto en cuestión.

La Figura 5-2 es una gráfica de dispersión, la cual es utilizada para representar la relación entre las variables (sensor Myoware, eje y, eje x y eje z del acelerómetro) para así identificar patrones, tendencias y correlaciones entre ellas:

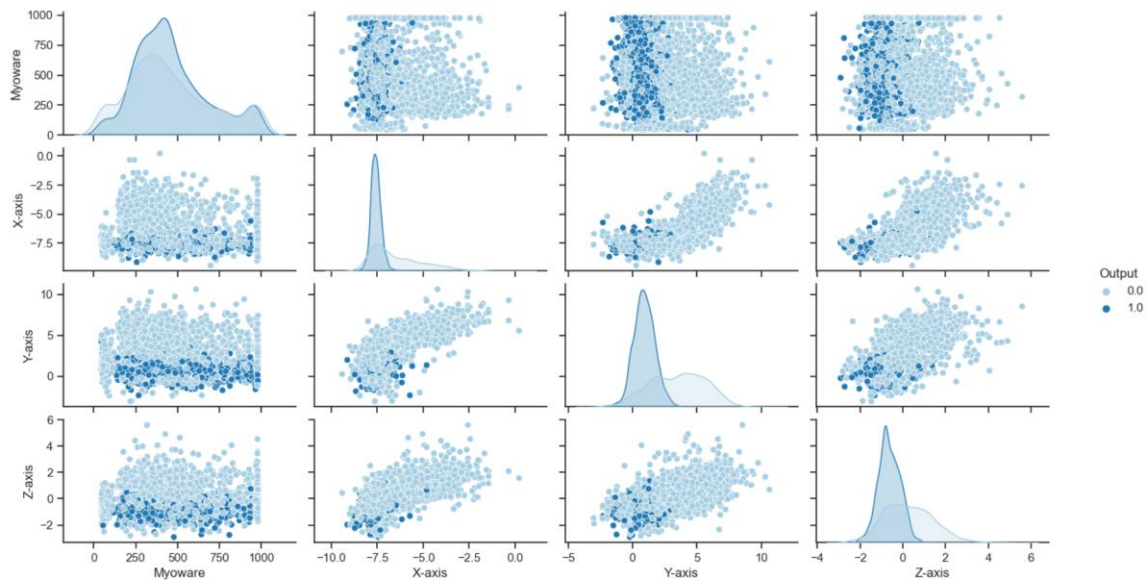


Figura 5-2 Gráfica de dispersión del Dataset

Con esta gráfica, se puede analizar la relación entre diversas características y su impacto en la correcta clasificación del movimiento de los bíceps. Como se mencionó previamente, la señal del sensor Myoware por sí sola no parece ser suficiente para distinguir entre posiciones correctas e incorrectas, dada la diversidad de características entre los sujetos de prueba. Sin embargo, la inclusión del acelerómetro permitió una diferenciación más precisa de la forma del levantamiento al generar patrones que ayudan a diferenciar los movimientos. Al combinar la característica de Myoware con alguna característica relacionada con los ejes del acelerómetro, se destaca más claramente la separación entre las dos clases, que se diferencian por su tonalidad en azul. También se observan valores atípicos, lo cual indica un comportamiento desfavorable para la implementación de algoritmos lineales, ya que se requieren regiones de decisión complejas para lograr una buena precisión en la clasificación de las dos clases.

Esta distinción se hace evidente en la Figura 5-3, donde se observa que ambas clases tienen valores muy similares si se considera únicamente la característica de los valores del sensor Myoware. En contraste, los valores del acelerómetro muestran una media de los datos de cada clase más divergente, lo que facilita una clasificación más exacta.

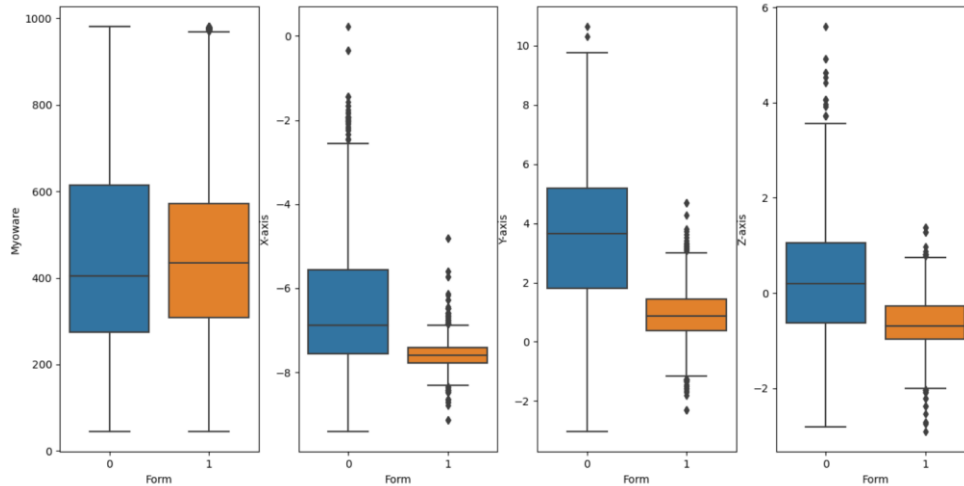


Figura 5-3 Boxplot del Dataset

Posteriormente, se procede a la implementación de los diversos algoritmos seleccionados. Para cada uno de ellos, se desarrolla un informe de clasificación, una matriz de confusión y se identifican las regiones de decisión. Estos análisis adicionales proporcionan una mayor claridad en la evaluación de los algoritmos y sus capacidades de clasificación. Para cada algoritmo se usó un 80% de los datos para entrenamiento y 20% para pruebas. Además, las entradas al algoritmo son: La señal EMG detectada por el sensor Myoware y los ejes x, y y z del acelerómetro. La salida de este es la clasificación si el movimiento fue correcto o incorrecto.

5.2.1. Algoritmo KNN

Para obtener la mayor exactitud, se entrena el algoritmo variando el valor de la cantidad de vecinos que se tiene en cuenta para el diagrama de fronteras para desarrollar una comparación entre los resultados. Después de pruebas, se encontró que con el valor $n_neighbors = 1$ se alcanza la mayor exactitud.

Tabla 5-1 Exactitud para distintos valores de $n_neighbors$

Número de Vecinos	Exactitud
KNN $n_neighbors = 1$	0,913
KNN $n_neighbors = 3$	0,871
KNN $n_neighbors = 5$	0,886
KNN $n_neighbors = 7$	0,89
KNN $n_neighbors = 9$	0,887
KNN $n_neighbors = 11$	0,883
KNN $n_neighbors = 13$	0,881
KNN $n_neighbors = 15$	0,883

El siguiente es el reporte de clasificación del algoritmo, en donde se muestra la exactitud del algoritmo y la precisión y sensibilidad de cada clase:

Tabla 5-2 Reporte de Clasificación KNN

Algoritmo KNN		
Exactitud	Precisión Movimiento Incorrecto	Precisión Movimiento Correcto
0,913	0,89	0,93

Se observa que la exactitud general fue del 91,3%. Individualmente, la clasificación de forma incorrecta de los bíceps tuvo una precisión del 89%, y que la clasificación de la forma correcta obtuvo una precisión del 93% por lo que se deduce que este algoritmo no presentó muchos falsos positivos.

5.2.2. Algoritmo Random Forest

El algoritmo Random Forest tiene diferentes parámetros, uno de ellos es “*criterion*”, el cual es una función para medir la calidad de una división en un árbol de decisión. Este parámetro tiene las opciones *gini*, *entropy* y *log_loss*, de los cuales *entropy* y *log_loss* presentan una exactitud mayor de %0,2 frente a *gini*.

El siguiente es el reporte de clasificación del algoritmo usando el criterio *entropy*:

Tabla 5-3 Reporte de Clasificación Random Forest

Algoritmo Random Forest		
Exactitud	Precisión Movimiento Incorrecto	Precisión Movimiento Correcto
0,941	0,94	0,94

Se consigue una buena exactitud general del 94,1%. Individualmente, en tanto la forma correcta e incorrecta del levantamiento se obtuvo una precisión del 94% por lo que se pudo evidenciar que hay muy pocos falsos positivos.

5.2.3. Algoritmo Naïve Bayes

El algoritmo Naïve Bayes tiene distintos métodos que pueden ser implementados por la librería SciKitLearn. Al emplear un reporte de exactitud se determinó que el de mayor exactitud es el *GaussianNB*.

Tabla 5-4 Exactitud para cada tipo de Naïve Bayes

Tipo	Exactitud
GaussianNB	0,834
BernoulliNB	0,827

ComplementNB 0,621
 MultinomialNB 0,642
 CategoricalNB 0,553

Se procede a desarrollar el reporte de clasificación al método Gaussiano el cual fue el que mejor exactitud obtuvo:

Tabla 5-5 Reporte de Clasificación Naïve Bayes

Algoritmo Naïve Bayes		
Exactitud	Precisión Movimiento Incorrecto	Precisión Movimiento Correcto
0,834	0,94	0,78

Para el algoritmo Naïve bayes se obtuvo una exactitud del 83,4%. Sin embargo, a pesar de que se obtuvo una precisión alta de 94% en la detección de movimientos incorrectos de los bíceps, también se consiguió una precisión menos favorable en la detección de movimientos correctos del 78%.

5.2.4. Algoritmo SVM

Para el algoritmo SVM, se usó un kernel tipo *rbf*, ya que a comparación de otros fue el que se determinó que tiene mayor precisión para este *dataset* gracias al reporte de exactitud que se empleó en cada tipo de kernel.

Tabla 5-6 Exactitud para distintos tipos de kernel de SVM

Kernel	Exactitud
rbf	0,88
linear	0,828
sigmoid	0,752
poly	0,787

También se notó imposible implementar un kernel *precomputed* debido a las características del *dataset* usado. Los demás parámetros no se cambiaron. Se procede a desarrollar el reporte de clasificación de dicho algoritmo:

Tabla 5-7 Reporte de Clasificación SVM

Algoritmo SVM		
Exactitud	Precisión Movimiento Incorrecto	Precisión Movimiento Correcto

0,88	0,92	0,85
------	------	------

Se logró una buena exactitud del 88%, una precisión del 92% en la clasificación de movimientos incorrectos y 85% en la clasificación de movimientos correctos.

5.2.5. Algoritmo MaxVoting

Se decidió implementar los modelos con mayor exactitud (Random Forest, KNN y SVM) para evaluar si se obtienen mejores resultados mediante este enfoque.

Según el reporte de clasificación se determinó que la exactitud de dicho modelo fue de 94,2%, lo cual solo fue un 0,1% más preciso que el algoritmo Random Forest.

Tabla 5-8 Reporte de Clasificación MaxVoting

Algoritmo MaxVoting		
Exactitud	Precisión Movimiento Incorrecto	Precisión Movimiento Correcto
0,942	0,95	0,93

6. Análisis de Resultados y Discusión

6.1. Análisis Resultados de los Algoritmos

Después de la implementación de todos los algoritmos, se determinó que el más preciso fue el MaxVoting, con una exactitud del 94,2%, seguido de cerca por el algoritmo Random Forest, con un 94,1%. En tercer lugar, se ubicó el algoritmo KNN, con una exactitud del 91,3%, seguido por el algoritmo SVM con kernel rbf, con un 88%. Por último, el algoritmo Naïve Bayes obtuvo una exactitud del 83,4%.

Tabla 6-1 Comparación entre el reporte de clasificación de los distintos algoritmos

	Naive Bayes	Random Forest	SVM	KNN	MaxVoting
Exactitud	83,4%	94,1%	88,0%	91,3%	94,2%
Precisión Mov. Correcto	78,0%	94,0%	85,0%	93,0%	95,0%
Precisión Mov. Incorrecto	94,0%	94,0%	92,0%	89,0%	93,0%
Falsos-Positivos	166	59	120	87	58

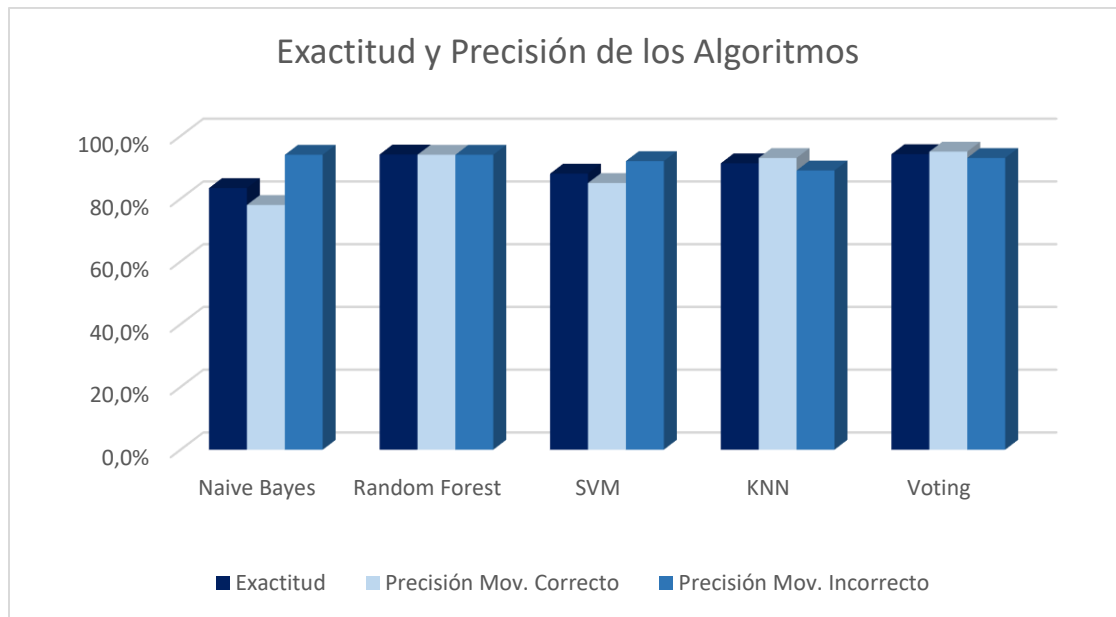


Figura 6-1 Gráfica de barras de exactitud y precisión de los algoritmos

Los dos algoritmos más precisos (MaxVoting y Random Forest) utilizan el método de *ensemble learning*, que mejora el desempeño predictivo al combinar las predicciones de múltiples modelos. Este método también es menos susceptible al sobreajuste y más estable ante cambios en los datos de entrenamiento.

Mediante el análisis de las matrices de confusión, se pudo concluir que el algoritmo de MaxVoting exhibió un falso-positivo menos en comparación con el algoritmo Random Forest, evidenciando así una mejora del 0,1%. Se espera que, al utilizar conjuntos de datos más extensos, esta mejora sea aún más notable.

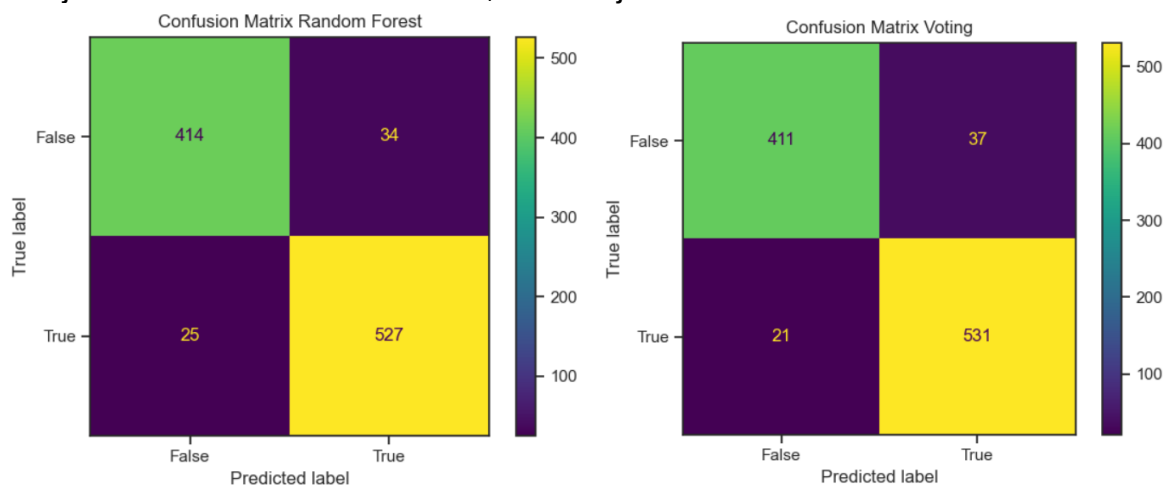


Figura 6-2 Matrices de confusión Random Forest y MaxVoting

En el diagrama de fronteras, se observa una similitud considerable en la clasificación realizada por ambos algoritmos. Sin embargo, al hacer zoom, se pueden apreciar con mayor claridad los factores que contribuyen a esta diferencia en la exactitud.

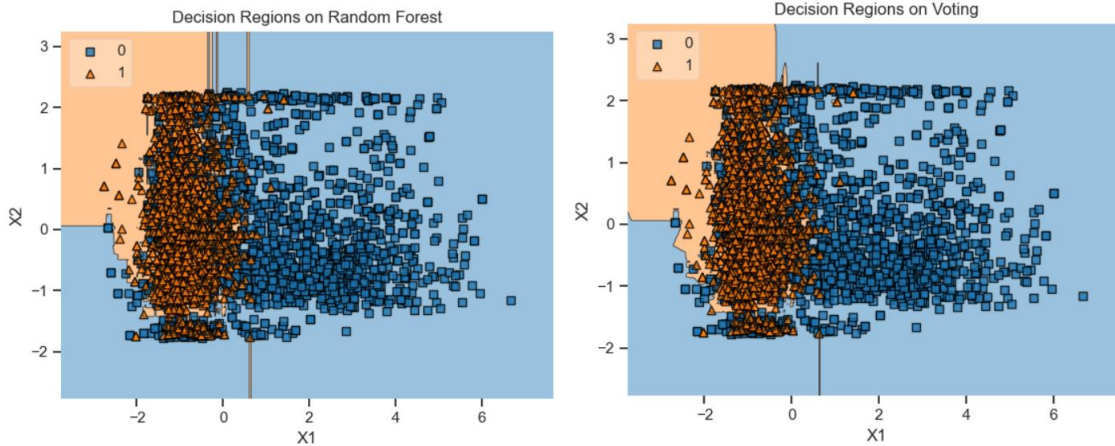


Figura 6-3 Diagramas de frontera de decisión Random Forest y MaxVoting

Esta disparidad en la exactitud surge debido al fenómeno del sobreajuste, el cual se manifiesta debido a la complejidad inherente del *dataset* y la presencia de ruido. MaxVoting, al ser un algoritmo de ensemble, amplía aún más la robustez contra el sobreajuste en comparación con el Random Forest individual, ya que integra las clasificaciones proporcionadas por los algoritmos Random Forest, KNN y SVM. En los diagramas de fronteras subsiguientes, se evidencia de manera más notable la presencia del sobreajuste, donde se aprecia que la frontera de clasificación generada por MaxVoting es más suave y precisa en comparación con Random Forest.

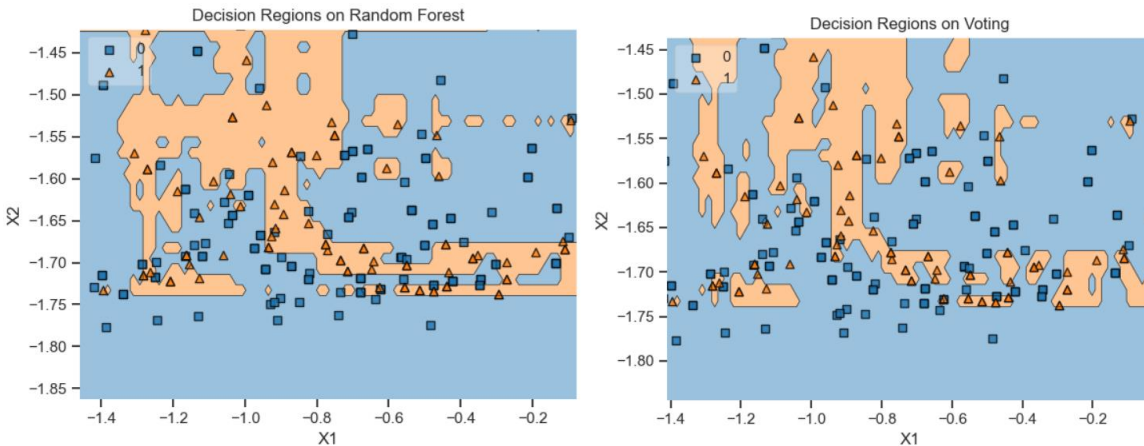


Figura 6-4 Diagramas de frontera de decisión Random Forest y MaxVoting con Zoom

El siguiente algoritmo es el algoritmo KNN el cual, aunque menos preciso que los anteriores, se destaca por su flexibilidad como método no paramétrico, capaz de manejar datos complejos y adaptarse a regiones de decisión no lineales sin asumir independencia entre las características del *dataset*. A partir de la matriz de confusión se sabe que el algoritmo obtuvo 87 falsos positivos, de los cuales el 57,4713% fueron en la clasificación de movimiento incorrecto.

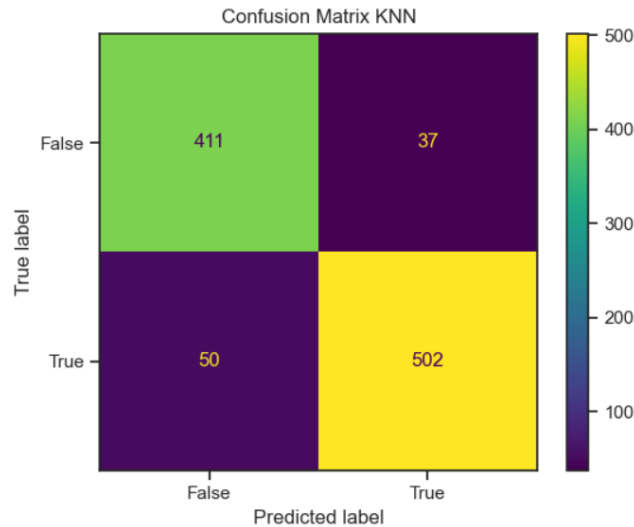


Figura 6-5 Matrices de confusión KNN

A continuación, se muestra el diagrama de fronteras que se obtuvo al desarrollar el algoritmo KNN:

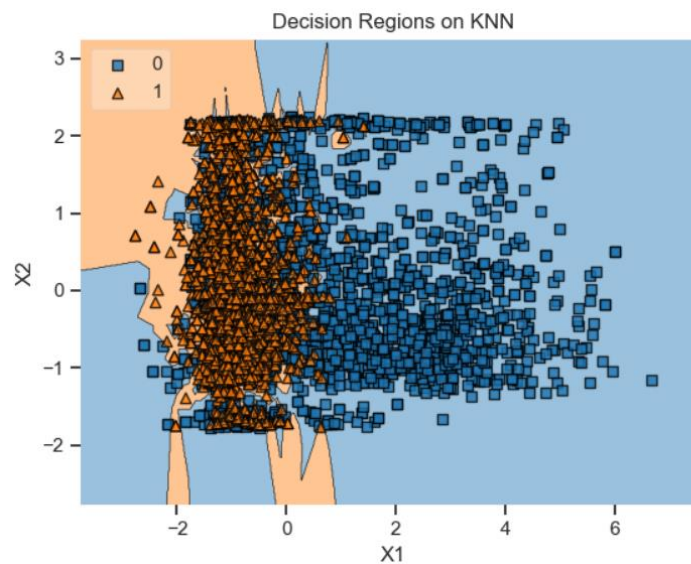


Figura 6-6 Diagramas de frontera de decisión KNN

Al aumentar el nivel de zoom en la imagen, se puede apreciar con precisión cómo este algoritmo ha logrado separar cada frontera de manera definida. Esto se debe principalmente a la elección de utilizar un único vecino, lo cual permite generar fronteras más precisas y complejas, aunque con la desventaja de requerir más recursos computacionales. Sin embargo, en la imagen de la derecha se observa una dificultad para clasificar algunos datos que se encuentran muy próximos entre sí, lo cual es una clara señal de sobreajuste. Esta dificultad puede atribuirse a las características complejas del *dataset*, donde la proximidad excesiva entre los puntos dificulta su clasificación. Es por esta razón que este algoritmo no alcanza la misma exactitud que los algoritmos anteriores.

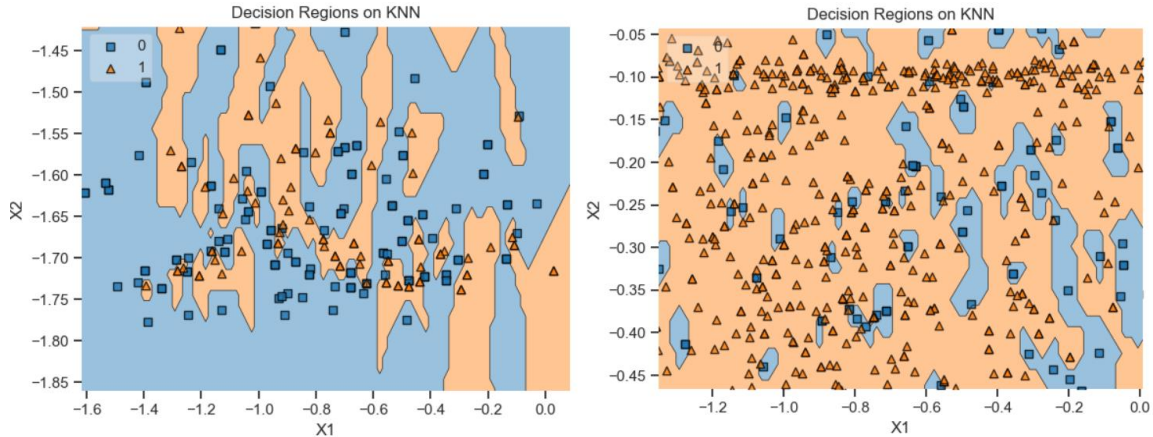


Figura 6-7 Diagramas de frontera de decisión KNN con Zoom

Aunque el algoritmo SVM mostró una buena exactitud, quedó detrás de los algoritmos mencionados previamente debido a su susceptibilidad al sobreajuste en *datasets* con numerosos *outliers*, como el implementado en este proyecto. La matriz de confusión evidencia esto ya que se obtuvieron 120 falsos-positivos, de los cuales el 75,8333% pertenecen a la clasificación de movimientos correctos de los bíceps.

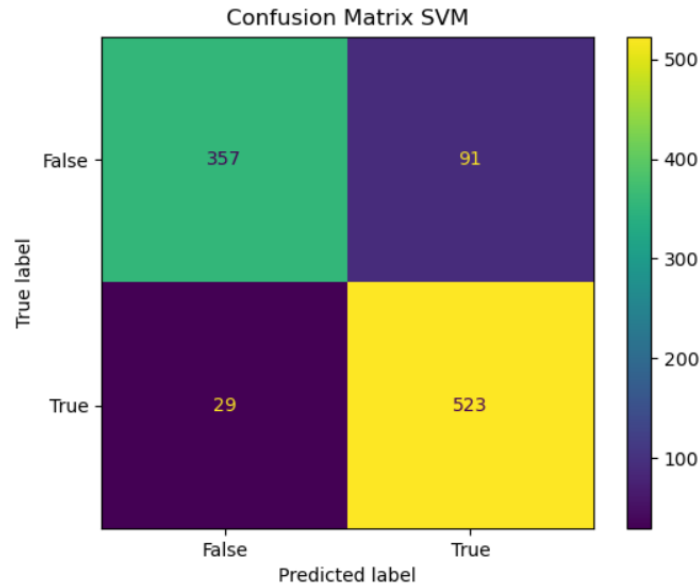


Figura 6-8 Matriz de Confusión SVM

El diagrama de fronteras de decisión ofrece una visión detallada de las clasificaciones erróneas realizadas por el algoritmo. Como se aprecia en la imagen izquierda, el algoritmo logró asignar correctamente la mayoría de los datos a su clase correspondiente. Sin embargo, al hacer zoom, como se muestra en la imagen derecha, aún se observa la presencia de ruido debido a la complejidad inherente del *dataset*.

Por lo tanto, se sugiere considerar el uso de un algoritmo no lineal más sofisticado para mejorar la precisión. A pesar de esto, el algoritmo actual demostró una exactitud bastante significativa. Además, dado que consume menos recursos

computacionales que los métodos previos, se perfila como una opción prometedora para la implementación en un exoesqueleto, gracias a su precisión, velocidad y simplicidad.

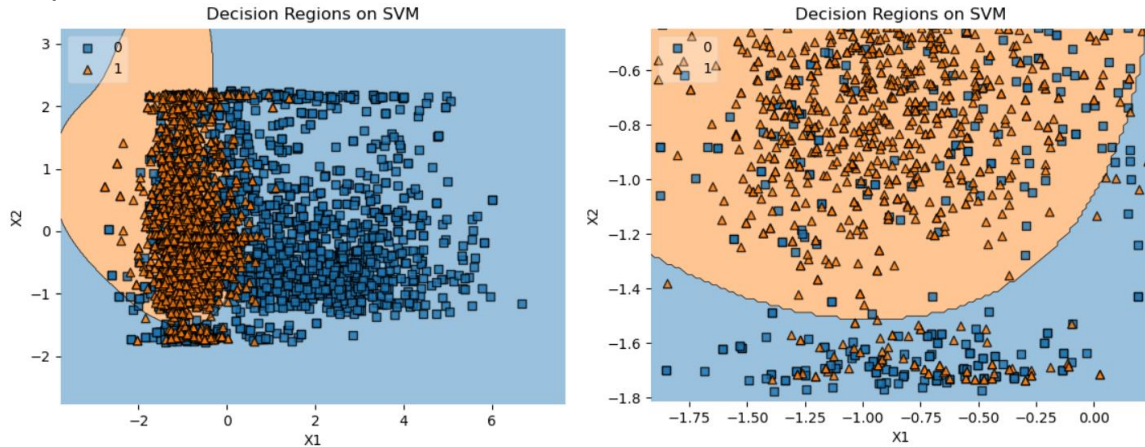


Figura 6-9 Diagrama de frontera de decisión SVM

Por último, se encuentra el algoritmo Naïve Bayes, que, aunque no lineal, logró una precisión del 83,4%, la cual es la más baja en comparación con los algoritmos previamente mencionados.

En la matriz de confusión informa que se obtuvieron una gran cantidad de falsos-positivos, en total 166, de los cuales la gran mayoría fue en la clasificación de movimientos correctos de los bíceps, con un 87,95181% de los falsos positivos.

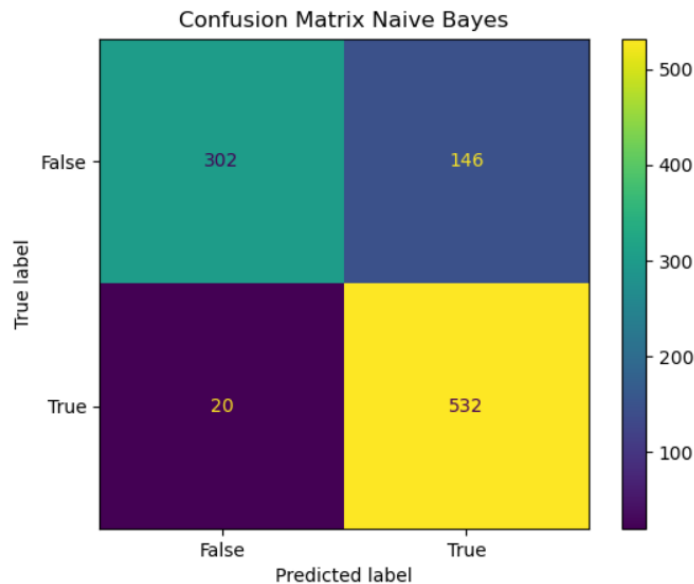


Figura 6-10 Matriz de Confusión Naïve Bayes

Los diagramas de frontera revelan una separación bastante simplista generada por este algoritmo. En la imagen de la derecha, se aprecia una clasificación con un alto nivel de ruido, lo que afecta la precisión general del modelo. Esta discrepancia se atribuye a la suposición de independencia entre las características empleada por el algoritmo, una premisa que no es ideal para este *dataset*, donde las características

están altamente correlacionadas. Esta situación conduce a resultados subóptimos, dado que la característica del Myoware depende en gran medida de los datos del acelerómetro para clasificar la calidad del movimiento.

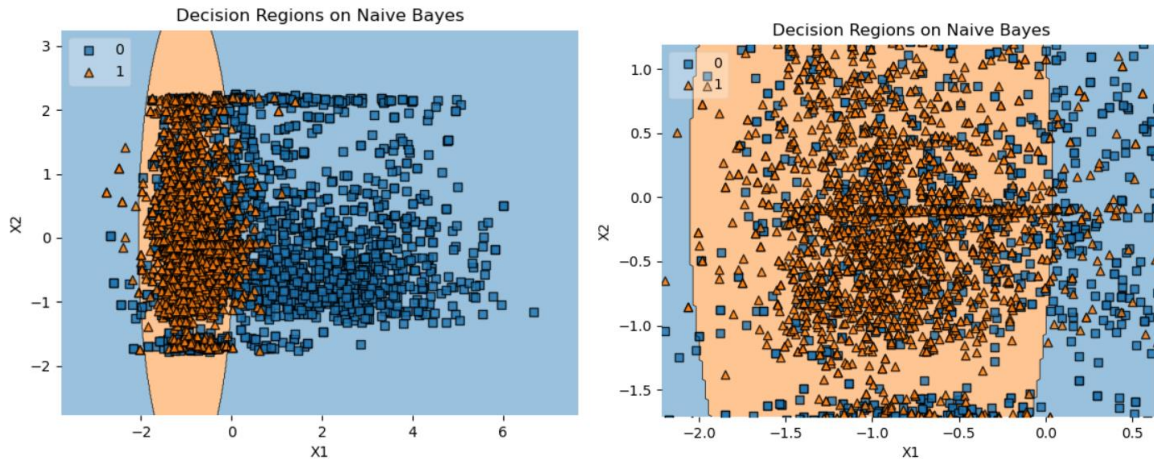


Figura 6-11 Diagrama de frontera de decisión Naïve Bayes

6.2. Diseño Estrategia de Control del Exoesqueleto

El diseño de la estrategia de control del exoesqueleto es un componente fundamental en el desarrollo de dispositivos de asistencia para la movilidad humana. Esta estrategia determina cómo el exoesqueleto interactúa con el usuario y cómo responde a las señales biológicas.

El objetivo principal de esta etapa del diseño es desarrollar una estrategia de control que permita al exoesqueleto proporcionar la asistencia necesaria para reducir la tensión muscular del usuario al desarrollar un movimiento incorrecto de los bíceps a partir de la clasificación del movimiento obtenida por el modelo *machine learning*.

Para el diseño del controlador, se toma como base el montaje utilizado por los investigadores en [33] para la recolección de datos. En este trabajo se ubicó el acelerómetro en el deltoide medio y el sensor de electromiografía en el centro de los bíceps. En la Figura 6-12 se puede apreciar claramente esta disposición:

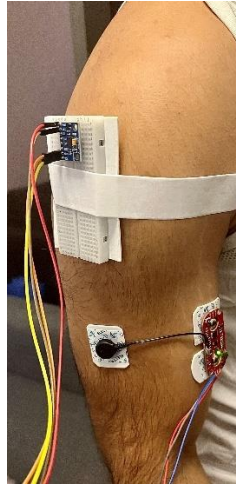


Figura 6-12 Montaje de la toma de datos [33]

Esta información es crucial para establecer la base del diseño del exoesqueleto, ya que se planea usar la misma posición del acelerómetro y sensor de electromiografía que se implementó en la toma de datos. Además de los sensores mencionados, se requiere un microcontrolador capaz de adquirir los datos y ejecutar el modelo de *machine learning* para detectar si se está realizando un levantamiento en una posición correcta.

6.2.1. Selección Microcontrolador e Implementación de Software

Para la adquisición de datos en la referencia [33], se utilizó un Arduino nano. Sin embargo, es esencial emplear un microcontrolador potente que pueda ejecutar el modelo mientras adquiere los datos. Por esta razón, se opta por el ESP-32 como la mejor opción, gracias a su asequibilidad y a sus características, como su procesador de 240 MHz de doble núcleo con 600 DMIPS de Tensilica LX6, lo que garantiza un mejor desempeño del sistema. Se elige el ESP-32 como microcontrolador para este proyecto, ya que otros como el Atmega328 tienen menos memoria y velocidad, lo que los hace menos adecuados para las necesidades del trabajo a realizar. Por otro lado, microcontroladores como el STM-32 ofrecen mayor estabilidad para implementaciones complejas a nivel industrial, pero su costo es significativamente más alto. Se considera que el ESP-32 proporciona un equilibrio óptimo entre rendimiento satisfactorio y un precio accesible.

Para la implementación del microcontrolador, es fundamental utilizar TensorFlow Lite, que permite exportar modelos de *machine learning* de Python a códigos en C++ que pueden ser interpretados por sistemas embebidos como los microcontroladores, sin requerir una gran cantidad de memoria y capacidad de procesamiento.

Otra alternativa más sencilla es el uso de la librería *Micro-Learn*, desarrollada por Adarsh Pal Singh y Sachin Chaudhari [34]. Esta librería permite implementar varios algoritmos de machine learning, como Perceptron, Logistic Regression, Gaussian Naïve Bayes (GNB), Passive-Aggressive Classifier (PA), Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA) y Support Vector Machine

(SVM) (Linear Kernel). Además, exporta los modelos programados en Python a códigos en C++ que son comprensibles para otros entornos, como ArduinoIDE.

Para ello se importa la librería de *Micro-Learn* y se carga el algoritmo empleando la función *Offload*, cuyo resultado es exportado a un archivo .ino con un código autogenerado el cual puede ser implementado eventualmente en un microcontrolador para desarrollar la clasificación de la calidad del movimiento de los bíceps.

Con el código, se puede controlar una variedad de actuadores utilizando microcontroladores compatibles con el entorno de desarrollo integrado (IDE) de Arduino. Estos actuadores tienen la función de generar la fuerza necesaria para mitigar la tensión muscular que podría resultar de un levantamiento incorrecto, particularmente en los bíceps. En la Figura 6-20 se presenta un diagrama de flujo representativo de la estructura del código.

6.2.2. Elementos de Hardware del Exoesqueleto

Se ha elegido utilizar dos servomotores dispuestos de tal manera que cada uno simule la funcionalidad de las dos cabezas de los bíceps (cabeza larga y cabeza corta, como se puede observar en la Figura 6-13). Esta configuración se logra montando los motores debajo de los trapezoides para evitar incomodidad a la hora del levantamiento. Los servomotores enrollarán un cable de manera análoga al mecanismo de una caña de pescar. Al tirar y enrollar los cables, se generará el movimiento de contracción en los bíceps, dado que dichos cables estarán conectados al antebrazo, simulando así la función de estos.



Figura 6-13 Ubicación de las cabezas de los bíceps [35]

El servomotor más potente se encargará de apoyar la cabeza larga de los bíceps, ya que esta es más susceptible a sufrir desgarros musculares, especialmente en la unión con el hombro. El cable unido a este servomotor estará unido cerca a la

muñeca para aprovechar al máximo su torque. Por otro lado, el servomotor más débil ayudará a la cabeza corta de los bíceps, proporcionando fuerza adicional y estabilidad durante el levantamiento.

Los servomotores estarán equipados con accesorios adicionales que tiran del cable para generar el movimiento deseado. No obstante, es importante considerar la distancia entre el borde del accesorio y el centro del rotor del motor, ya que esto afecta la fuerza efectiva que el motor puede proporcionar. A continuación, se presenta la fórmula para calcular la masa que el servomotor realmente puede levantar, donde t representa el torque nominal del motor y l representa la longitud del brazo, en este caso, la longitud del accesorio que está unido al servomotor:

$$masa\ a\ levantar = \frac{t}{l} \quad (9)$$

Para este proyecto, se ha optado por utilizar servomotores DS-R026 y RDS5160 con capacidades de 75 kg*cm y 60 kg*cm respectivamente para proporcionar la fuerza requerida para levantamientos pesados. Teniendo en cuenta que el accesorio tiene un radio de 2 cm, se calcula que la masa que cada servomotor puede levantar es de 37,5 kg y 30 kg, respectivamente.

Con el microcontrolador, es posible regular el voltaje suministrado a los servomotores para ajustar el torque según las necesidades específicas. Esto incluye la capacidad de disminuir o aumentar el torque del motor en función de la señal detectada por el sensor EMG.

El sensor EMG se posiciona en el centro de los bíceps para una detección precisa de la señal, siguiendo la recomendación del fabricante. A continuación, se muestra una imagen explicativa obtenida del *datasheet* del sensor:

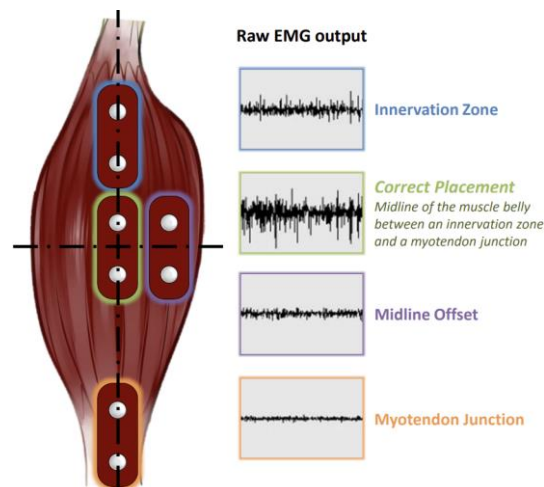


Figura 6-14 Posicionamiento Sensor EMG [36]

Para asegurar la mayor exactitud, se ubica el acelerómetro en la misma posición en la que se desarrolló la toma de datos para el *dataset*, la cual fue en el deltoide medio. Con esta información en cuenta se procede a desarrollar el diseño del exosqueleto el cual se presenta en la Figura 6-15:

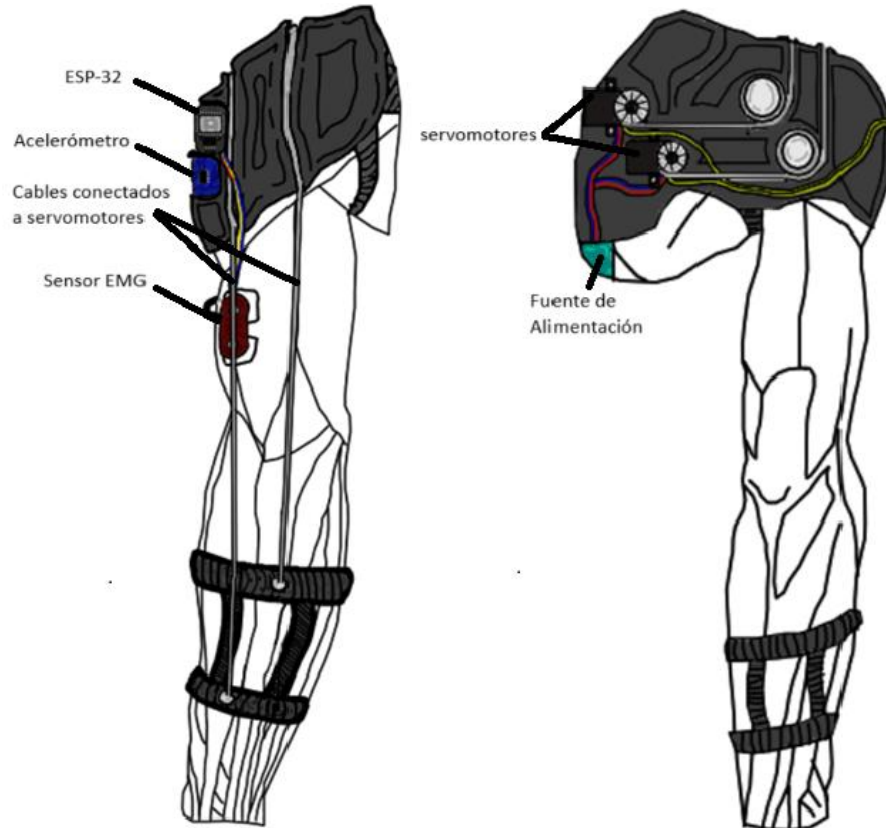


Figura 6-15 Dibujo Exoesqueleto vista frontal y trasera desarrollada en Krita

6.2.3. Propuesta de la Estrategia de Control

En general, el control del exoesqueleto se divide en dos etapas principales. Primero, está el desarrollo del modelo utilizando algoritmos programados en Python. Durante esta fase, se procesa el conjunto de datos seleccionado y se procede con el entrenamiento del modelo resultante. Posteriormente, se realiza una validación de la clasificación ejecutada por el algoritmo (la cual se evidencia en la sección 5.2), la cual se utiliza para mejorar los resultados hasta alcanzar la mejor predicción posible, considerando las capacidades y limitaciones del algoritmo seleccionado.

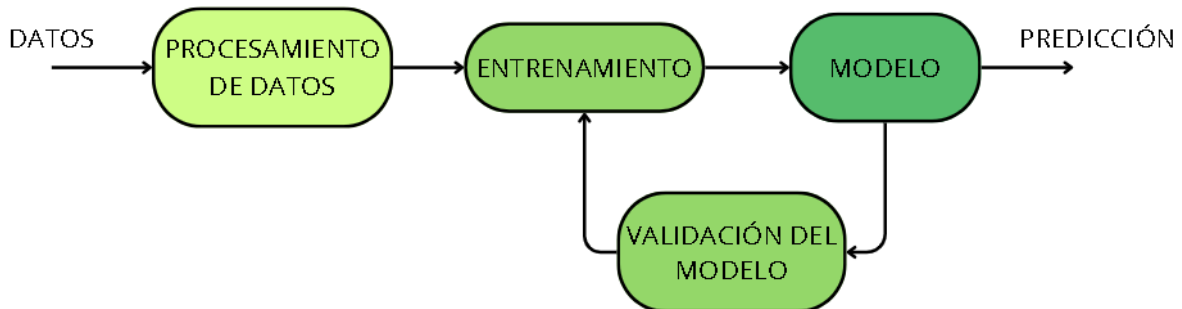


Figura 6-16 Diagrama del Funcionamiento del Algoritmo

Luego, entra en juego la etapa de control del exoesqueleto a través del microcontrolador, que ya cuenta con el modelo de clasificación de la calidad del movimiento de los bíceps obtenida en el paso anterior y actúa a partir de esta para reducir la tensión muscular que puede enfrentar el músculo. En la Figura 6-17, se representa de forma gráfica el funcionamiento del exoesqueleto en general.

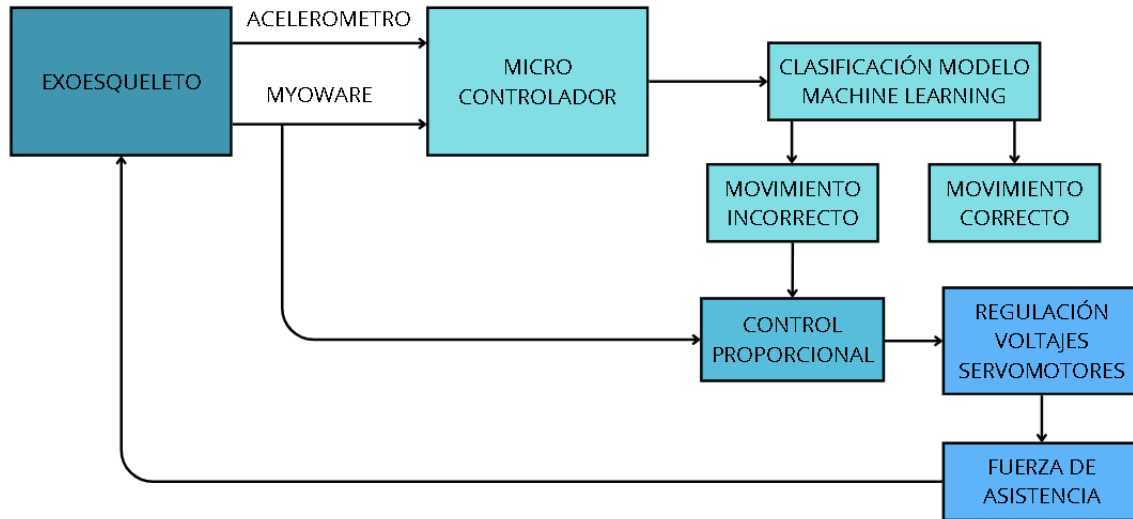


Figura 6-17 Diagrama del Funcionamiento del Exoesqueleto

En primer lugar, se procede con la lectura de los datos provenientes del sensor de electromiografía y del acelerómetro. Posteriormente, estos datos son analizados por el modelo para generar una predicción sobre el tipo de movimiento realizado. Si el movimiento detectado puede presentar un riesgo a la integridad de los bíceps, se procede a activar los servomotores. Además, se implementa un control proporcional de lazo cerrado, el cual ajusta el valor de voltaje de entrada de los servomotores proporcional a la activación muscular detectada por el sensor EMG.

También se consideran los valores del acelerómetro para determinar la posición del brazo, ya que esta afecta la distribución del esfuerzo en las dos cabezas de los bíceps. Cuando el levantamiento se realiza con el antebrazo apuntando hacia afuera, se ejerce una mayor carga sobre la cabeza corta de los bíceps. Por otro lado, cuando el antebrazo apunta hacia adentro, se genera una mayor tensión en la cabeza larga de los bíceps. Si aumenta el riesgo de que la integridad de una de las cabezas de los bíceps se vea comprometida, se aumenta el voltaje de entrada del servomotor correspondiente para que este entregue mayor torque y así reducir la tensión muscular.

En la Figura 6-18 se presentan los levantamientos mencionados junto con su movimiento en los ejes, tal como registra el acelerómetro. Se aprecia que, cuando el valor de x aumenta, se manifiesta un movimiento en el que la mayor parte del

esfuerzo recae en la cabeza larga de los bíceps. Por otro lado, cuando el valor de x disminuye, la carga se distribuye principalmente hacia la cabeza corta de los bíceps.

Gracias al dataset, se sabe que el valor promedio de x cuando el movimiento es correcto es de -7,62. Por lo tanto, se ajusta este valor en el microcontrolador para que sea el nuevo punto de origen, de manera que en este punto se tenga un valor de 0. Esto permite identificar cuál cabeza del bíceps está siendo sobre-esforzada a partir del signo de la variable x : si x es positivo, se está sobre-esforzando la cabeza larga de los bíceps por lo que se aumenta el torque del servomotor 1; si x es negativo, se está sobre-esforzando la cabeza corta de los bíceps por lo que se aumenta el torque del servomotor 2.

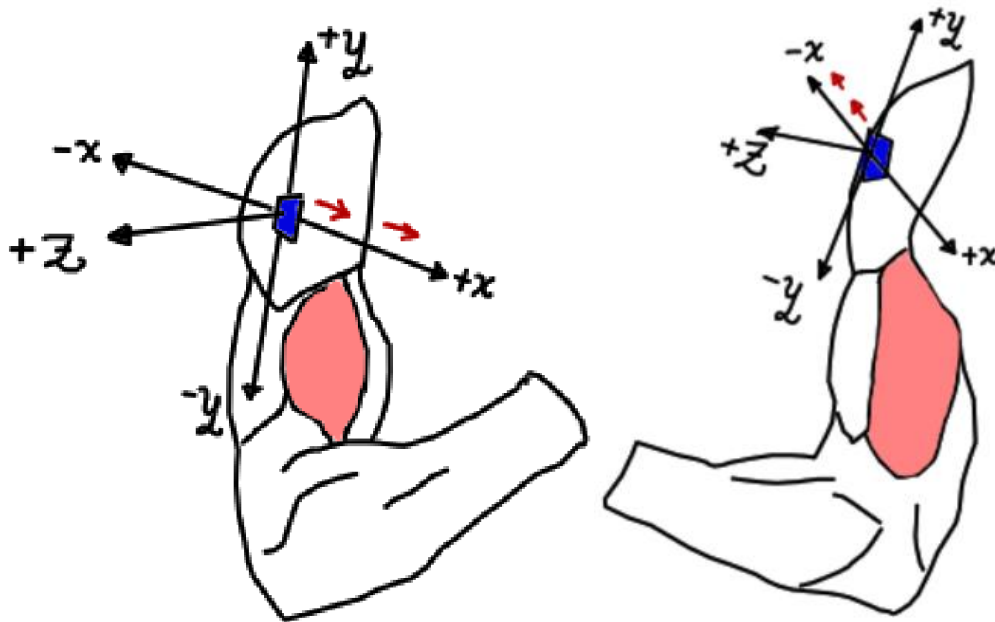


Figura 6-18 Movimientos que causan más esfuerzo en las cabezas de los bíceps

Con toda esta información, se puede diseñar el funcionamiento del control proporcional de lazo cerrado que permita controlar la fuerza entregada por los servomotores para reducir la tensión muscular que ejercen los bíceps. A continuación, en la Figura 6-19, se presenta el diagrama de bloques de dicho control:

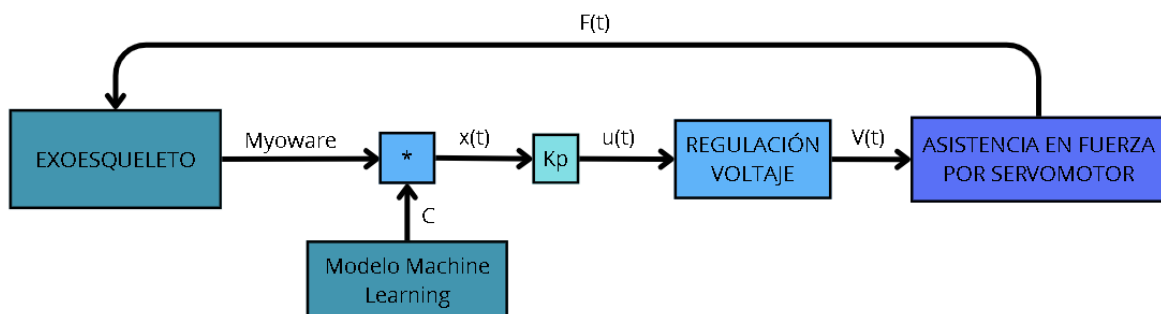


Figura 6-19 Diagrama del Bloques del Control Proporcional de Lazo Cerrado

En este sistema, se considera la amplitud de la señal EMG obtenida por el sensor Myoware y un valor C que representa el movimiento detectado por el algoritmo. Estos dos valores son multiplicados para así otorgar la asistencia o no, dependiendo del valor de C : La variable C tiene un valor de 1 o 0, cuando se identifica un movimiento incorrecto el valor de C es 1, por lo que se desarrolla el control proporcional para así otorgar la asistencia necesaria; Cuando se identifica un movimiento correcto, el valor de C es 0, por lo que ya no entra ninguna señal a la etapa de regulación de voltaje y no se otorga ninguna asistencia. Después de determinar si se necesita la asistencia, se procede a implementar un control proporcional para aumentar o disminuir el nivel de fuerza de asistencia entregada por el exoesqueleto.

Se procede a establecer como se calcula el valor de la variable de control proporcional Kp . Para iniciar, se procede a desarrollar la ecuación que representa el control empleado en el microcontrolador:

$$u(t) = Kp * Myoware * C \quad (10)$$

Se puede deducir que entre mayor sea el valor de la amplitud de la señal EMG, mayor será la asistencia que otorgarán los servomotores. Sin embargo, dependiendo de la posición del brazo, se debe otorgar mayor asistencia a alguna de las dos cabezas de los bíceps por lo que para el cálculo de Kp se debe tener en cuenta el eje x del acelerómetro.

$$Kp = 1 + \left(\frac{|x|}{10} \right) \quad (11)$$

A partir del dataset, se puede observar que el valor absoluto máximo que alcanza es 8, por lo que se decide dividir el valor absoluto de esta variable por 10, con la finalidad de obtener un valor entre 0 y 1. Al sumar este valor con 1, se logra obtener un valor de una ganancia Kp entre 1 y 2, la cual permitirá aumentar el valor de la señal proporcionalmente al valor absoluto de x . Entre mayor sea el valor de x , mayor debe ser la asistencia otorgada por el exoesqueleto para aliviar la tensión muscular ya que se está efectuando un movimiento que sobre-esfuerza una de las cabezas de los bíceps.

Este valor de la señal resultante $u(t)$, se empleará posteriormente para determinar la alimentación del servomotor, con el fin de ajustar su torque mediante la regulación del voltaje. Es importante aclarar que este valor se aplica específicamente al servomotor que asiste a la cabeza que está experimentando sobre-esfuerzo, este servomotor se elige a partir del signo de x , como se había especificado previamente. Para el otro servomotor, se utiliza el valor $Kp = 1$.

A continuación, se presenta un diagrama de flujo el cual representa el funcionamiento del código del microcontrolador el cual emplea el control

anteriormente explicado para así aliviar la tensión en las respectivas cabezas de los bíceps:

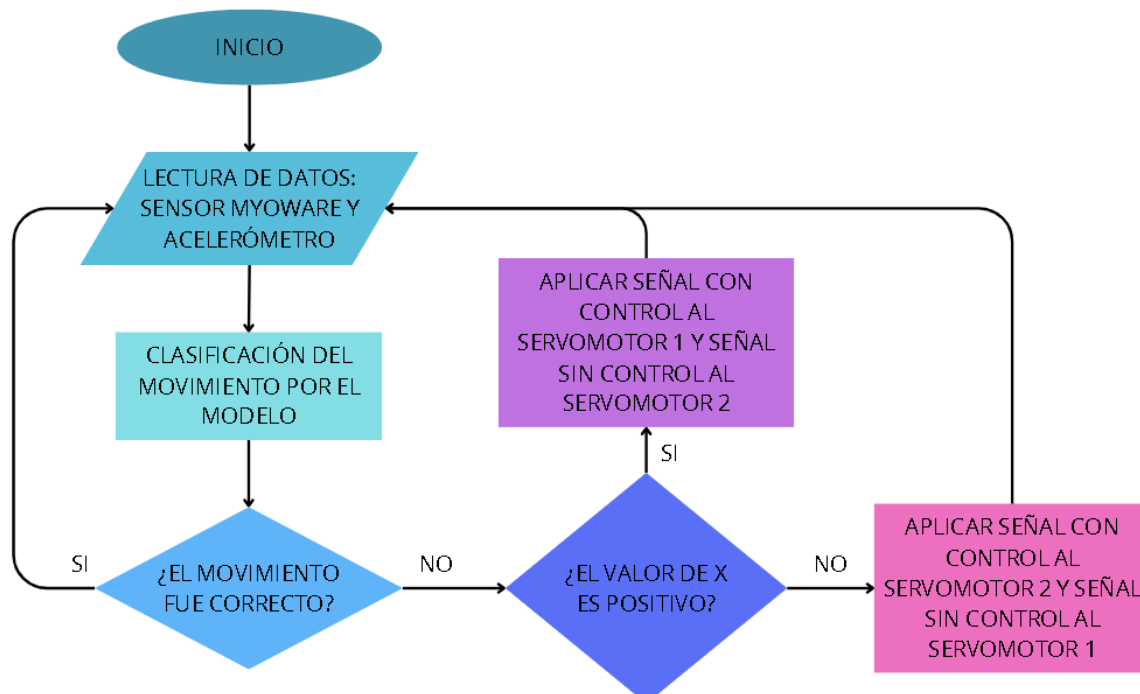


Figura 6-20 Diagrama de Flujo del Código

Para validar el funcionamiento del control, se desarrolla una simulación en Matlab donde se aplica el control descrito previamente a una señal EMG. Dicha simulación se presenta en la Figura 6-21. La gráfica se divide en distintas secciones, en las que los valores de x y del sensor Myoware se modifican para cambiar la señal de entrada, con el objetivo de aumentar el torque del servomotor y aliviar la tensión en los bíceps proporcionalmente al esfuerzo detectado y la posición del brazo.

En la primera sección, el valor de C es 0, representando un movimiento correcto donde ambas cabezas de los bíceps ejercen fuerza sin sobre esforzarse, por lo que la señal de salida es 0 y no se ejerce ninguna asistencia en el levantamiento. En las próximas secciones, C posee un valor de 1, por lo que se identifica un movimiento incorrecto y se debe entregar una asistencia. La segunda, tercera y cuarta sección tienen valores de x de 4, 2 y 1 respectivamente, representando posiciones que incrementan ligeramente la tensión en una de las cabezas de los bíceps, por lo que se aumenta levemente el torque del servomotor encargado de aliviar la tensión en esa cabeza. Finalmente, en la última sección, x tiene un valor de 8, lo cual representa una posición que causa una gran tensión en una de las cabezas de los bíceps, requiriendo un aumento considerable del torque aplicado por el servomotor para evitar la sobretensión en dicha cabeza.

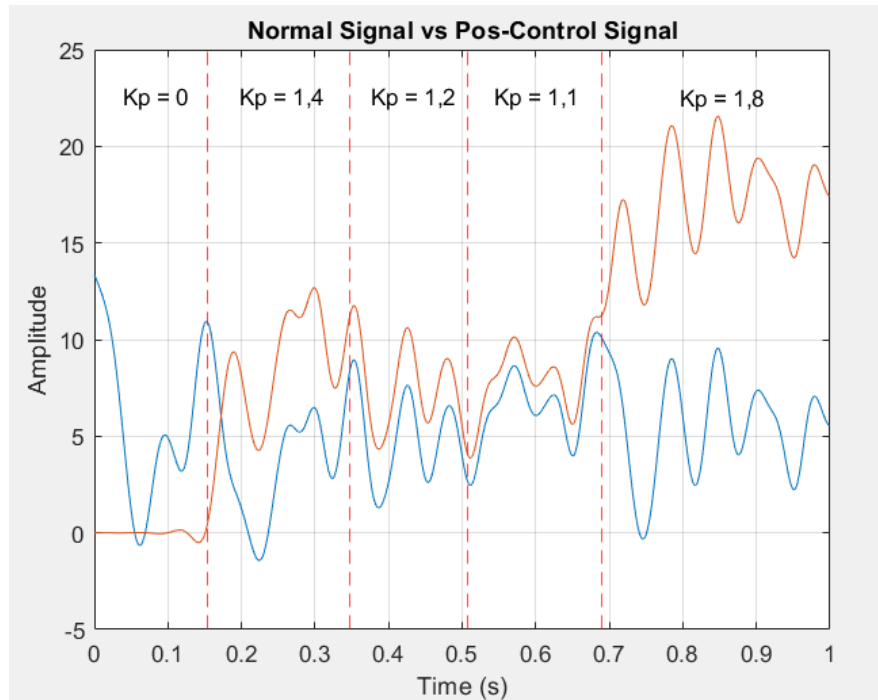


Figura 6-21 Simulación del Control Proporcional

Esta modificación del torque de cada servomotor se logra ajustando el voltaje de entrada, el cual se controla proporcionalmente al valor de la señal después de aplicar el control. Cuanto mayor sea la amplitud de la señal, mayor será el voltaje proporcionado, lo que incrementará el torque entregado por el servomotor.

Este ajuste del voltaje se realiza a través de un regulador de voltaje variable, permitiendo así modificar el torque de los servomotores. Este proceso de modificación del voltaje se lleva a cabo mediante el control proporcional del valor de resistencia de un potenciómetro digital con respecto a los valores de la señal *Data* obtenidos en el control previo.

Para la regulación de voltaje, se emplea un LM350, un dispositivo capaz de ajustar el voltaje de salida mediante un potenciómetro. La elección de este regulador se fundamenta en su capacidad para variar el voltaje de salida entre 1,2V y 33V con una alta precisión, además de su capacidad para tolerar corrientes de hasta 3A, asegurando así que pueda manejar la corriente requerida por los servomotores. En caso de necesidad, es posible aumentar el límite de corriente hasta 10A mediante la inclusión de un transistor en el circuito. A continuación, en la Figura 6-22, se muestra el circuito proporcionado por el *datasheet*:

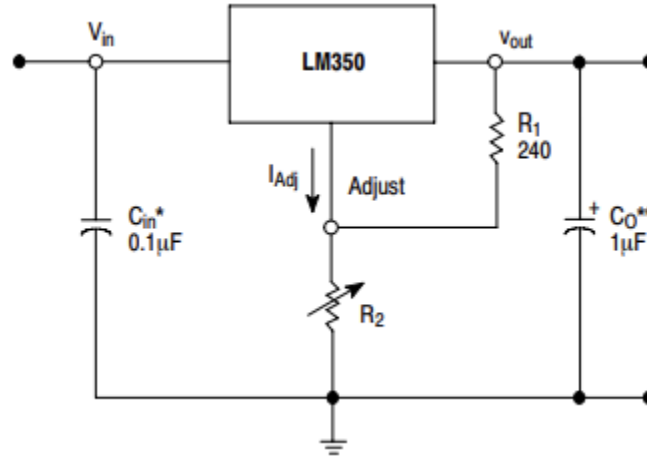


Figura 6-22 Diagrama de conexiones del LM350 [37]

La ecuación otorgada por el *datasheet* para calcular el valor del voltaje de salida a partir de la resistencia y el potenciómetro es la siguiente:

$$V_{out} = 1,25V * \left(1 + \frac{R_2}{R_1}\right) + I_{ADJ} * R_2 \quad (11)$$

Por lo general, en este cálculo se suele ignorar el valor de I_{ADJ} debido a que es aproximadamente igual a 50µA. Para poder controlar esta regulación de voltaje mediante el microcontrolador se usa un potenciómetro digital que estará ubicado en lugar de R_2 , con el fin de controlar el voltaje de salida del LM350 al cambiar el valor de resistencia de manera proporcional a los valores de $u(t)$ obtenidos previamente con los valores del sensor Myoware y el acelerómetro. El potenciómetro digital seleccionado es el X9C102S de 1000Ω, el cual garantiza, junto una R_1 de 204Ω, que el valor máximo de voltaje en la salida sea de 7,37V, ya que ambos servomotores mencionados trabajan con su máximo torque a 7,4V en su alimentación. Si se necesita un voltaje de salida mayor, se puede optar por cambiar el potenciómetro a X9C103S, X9C503S o X9C104S, los cuales tienen valores de resistencia de 10kΩ, 50kΩ y 100kΩ respectivamente. También se puede ajustar la resistencia R_1 para obtener el valor de salida deseado según el servomotor que se vaya a utilizar.

Por último, para garantizar la seguridad del usuario, se incorpora un botón en el dedo medio de la mano, diseñado para ser fácilmente accionado con los dedos índice o anular. Su propósito es desactivar los servomotores del exoesqueleto en caso de que la persona desee detener su funcionamiento.

Cabe recalcar, que el tiempo de respuesta del exoesqueleto esta determinado por la velocidad en la efectuación de la clasificación de los movimientos y el tiempo de ajuste del regulador de tensión. Según el *datasheet* del LM350, este tiempo de ajuste entre cambios de voltaje es alrededor de 10µs, por lo que este tiempo no afectará notoriamente el tiempo de respuesta del control en sí.

A continuación, en la Figura 6-23, se muestra las conexiones entre los componentes y el microcontrolador que garantiza el funcionamiento del control de la regulación del voltaje de uno de los servomotores:

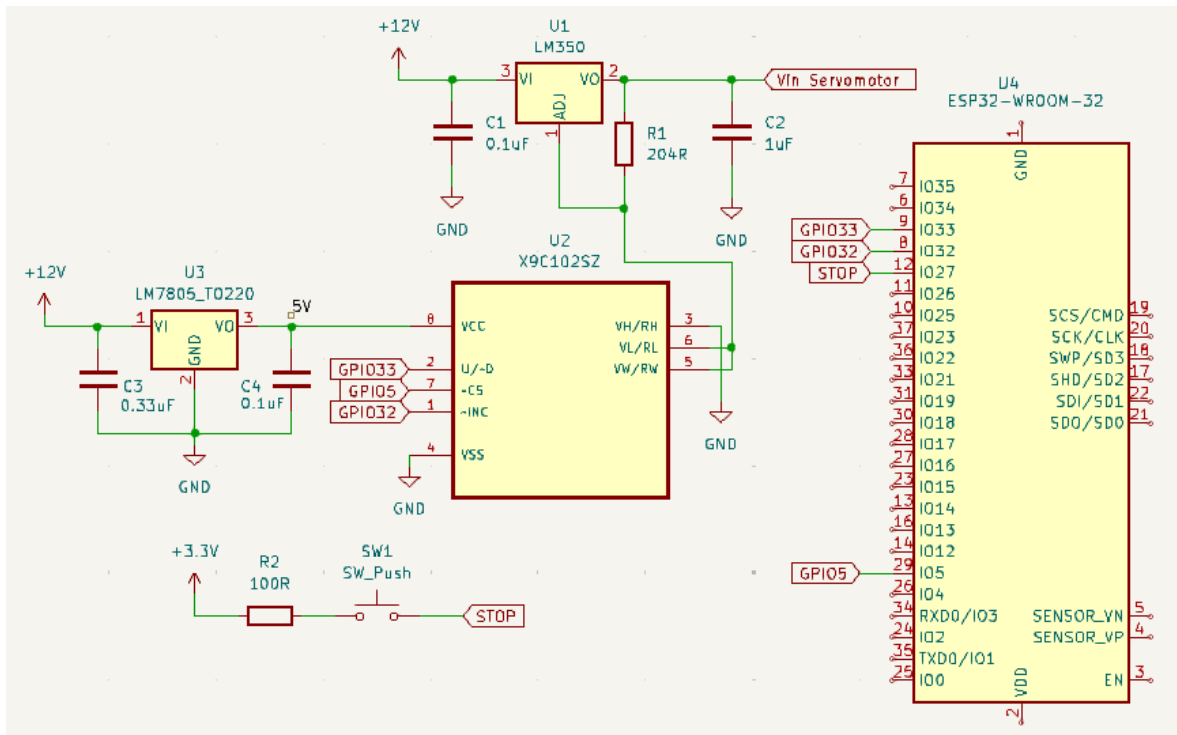


Figura 6-23 Circuito del Control de la Regulación de Voltaje

Con todo lo anteriormente explicado, se sientan las bases teóricas para el diseño tanto del controlador como del exoesqueleto. Ambos componentes están diseñados para implementar un modelo de clasificación de *machine learning* que permita aliviar la tensión muscular de los bíceps. Este enfoque integrado entre el hardware del exoesqueleto y el modelo de *machine learning* es fundamental para lograr una asistencia adecuada y personalizada para el usuario con respecto al esfuerzo realizado sus bíceps, adaptándose dinámicamente a sus necesidades individuales y proporcionando una experiencia de uso cómoda y eficiente.

7. Conclusiones y Trabajos futuros

Los algoritmos clasificatorios seleccionados han exhibido resultados prometedores, con una exactitud superior al 80%. Empleando el sensor Myoware y el acelerómetro, dichos algoritmos lograron discernir entre movimientos correctos e incorrectos de los bíceps. Es relevante resaltar que los algoritmos no lineales demostraron gran exactitud, atribuible a su capacidad para modelar relaciones no lineales y ofrecer mayor robustez frente a datos atípicos.

El Random Forest destacó como el segundo algoritmo más preciso, alcanzando una exactitud del 94.1%. Esta eficacia se fundamenta en su capacidad para generar múltiples árboles de decisión y promediar sus predicciones, lo que reduce el sesgo y la varianza del modelo, mitigando así el sobreajuste.

En relación con el algoritmo MaxVoting, se implementaron los tres algoritmos más precisos (Random Forest, KNN y SVM), logrando una leve mejora respecto al Random Forest individual. Esta técnica de ensamble aprovecha las predicciones de múltiples algoritmos para corregir posibles errores individuales y mejorar la generalización del modelo. No obstante, es relevante señalar que su implementación requiere una mayor carga computacional, lo que podría no ser ideal en aplicaciones que demandan tiempos de respuesta rápidos, como la implementación en un exoesqueleto.

Para futuras investigaciones, se sugiere explorar algoritmos con menor complejidad computacional, como SVM, que podrían ser más adecuados para aplicaciones en tiempo real, como el control de exoesqueletos. Asimismo, se plantea la posibilidad de extender el alcance de este estudio a otros grupos musculares y movimientos, aprovechando la versatilidad del sensor Myoware y el acelerómetro. Este enfoque ampliaría las oportunidades de aplicación y permitiría abordar una variedad de problemas relacionados con el control de exoesqueletos en diversos contextos. En conclusión, este proyecto ha establecido una base sólida para futuras investigaciones en el campo del control de exoesqueletos, ofreciendo una plataforma flexible y adaptable a diferentes necesidades y objetivos de investigación.

8. Referencias Bibliográficas

- [1] P. A. R. López, «Universidad Pontificia Bolivariana», "El riesgo de las lesiones musculares en el trabajo", UPB. <https://www.upb.edu.co/es/noticias/lesiones-musculares-riesgos-entabajadores> (accedido el 18 de agosto de 2023)<http://www.scielo.org.co/pdf/recis/v16n2/1692-7273-recis-16-02-203.pdf> (accedido el 18 de agosto de 2023).
- [2] SciELO Colombia- Scientific Electronic Library Online. <http://www.scielo.org.co/pdf/recis/v16n2/1697273-recis-16-02-203.pdf> (accedido el 18 de agosto de 2023).
- [3] <https://www.mintrabajo.gov.co/mintrabajo-llama-la-atencion-al-sector-de-la-construccionpara-implementar-medidas-que-permitan-bajar-indices-de-accidentalidad> (accedido el 18 de agosto de 2023).
- [4] Noack Law Office. "Las lesiones más comunes de la industria manufacturera | Noack Law Office". Noack Law Office. <https://www.noacklawoffice.com> (accedido el 18 de agosto de 2023).
- [5] «Tendinitis - Síntomas y causas - Mayo Clinic», Top-ranked Hospital in the Nation – Mayo Clinic, 2023. dirección: <https://www.mayoclinic.org/es/diseases-conditions/tendinitis/symptoms-causes/syc-20378243>.
- [6] E. Traumatología, «Prevención y rehabilitación de los desgarros musculares», Elgeadi Traumatología, dirección: <https://elgeaditraumatologia.com/prevencion-rehabilitacion-desgarro-muscular/>.
- [7] S. S. Rasheed e I. H. Glob, «Classifying and Prediction for Patient Disease Using Machine Learning Algorithms», en 2022 3rd Information Technology To Enhance e-learning and Other Application (IT-ELA), 2022, págs. 196-200. DOI: 10.1109/IT-ELA57378.2022. 10107935.
- [8] N. Unidas, «Objetivos de Desarrollo Sostenible», Naciones Unidas, 2023. dirección: <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollosostenible/>. 18 19
- [9] «En febrero de 2023, el empleo en la construcción creció 5,1», Observatorio de Desarrollo Económico, 2023. dirección: <https://observatorio.desarrolloeconomico.gov.co/construccion-bogota/en-febrero-de-2023-el-empleo-en-laconstruccion-crecio-51#:~:text=El%20n%C3%BAmero%20de%20personas%20ocupadas,mil%20personas%20ocupadas%20en%202023..>

- [10] G. Li, D. Bai, G. Jiang et al., «Continuous dynamic gesture recognition using surface EMG signals based on blockchain-enabled internet of medical things», *Information Sciences*, vol. 646, pág. 119 409, 2023, ISSN: 0020-0255. DOI: <https://doi.org/10.1016/j.ins.2023.119409>. dirección: <https://www.sciencedirect.com/science/article/pii/S0020025523009945>.
- [11] L. Wang, J. Fu, H. Chen y B. Zheng, «Hand gesture recognition using smooth wavelet packet transformation and hybrid CNN based on surface EMG and accelerometer signal», *Biomedical Signal Processing and Control*, vol. 86, pág. 105 141, 2023, ISSN: 1746-8094. DOI: <https://doi.org/10.1016/j.bspc.2023.105141>. dirección: <https://www.sciencedirect.com/science/article/pii/S1746809423005748>.
- [12] K. H. Lee, J. Y. Min y S. Byun, «Electromyogram-Based Classification of Hand and Finger Gestures Using Artificial Neural Networks», *Sensors*, vol. 22, n.o 1, 2022, ISSN: 1424-8220. DOI: 10.3390/s22010225. dirección: <https://www.mdpi.com/1424-8220/22/1/225>.
- [13] V. Khodadadi, F. Nowshiravan Rahatabad, A. Sheikhan y N. Jafarnia Dabanloo, «Nonlinear analysis of biceps surface EMG signals for chaotic approaches», *Chaos, Solitons Fractals*, vol. 166, pág. 112 965, 2023, ISSN: 0960-0779. DOI: <https://doi.org/10.1016/j.chaos.2022.112965>. dirección: <https://www.sciencedirect.com/science/article/pii/S0960077922011444>.
- [14] P. Xiong, C. Wu, H. Zhou, A. Song, L. Hu y X. P. Liu, «Design of an accurate end-of-arm force display system based on wearable arm gesture sensors and EMG sensors», *Information Fusion*, vol. 39, págs. 178-185, 2018, ISSN: 1566-2535. DOI: <https://doi.org/10.1016/j.inffus.2017.04.009>. dirección: <https://www.sciencedirect.com/science/article/pii/S1566253517302695>.
- [15] M. Barfi, H. Karami, F. Faridi, Z. Sohrabi y M. Hosseini, «Improving robotic hand control via adaptive Fuzzy-PI controller using classification of EMG signals», *Heliyon*, vol. 8, n.o 12, e11931, 2022, ISSN: 2405-8440. DOI: <https://doi.org/10.1016/j.heliyon.2022.e11931>. dirección: <https://www.sciencedirect.com/science/article/pii/S2405844022032194>. 20
- [16] B. Chen, Y. Zhou, C. Chen et al., «Volitional control of upper-limb exoskeleton empowered by EMG sensors and machine learning computing», *Array*, vol. 17, pág. 100 277, 2023, ISSN: 2590-0056. DOI: <https://doi.org/10.1016/j.array.2023.100277>. dirección: <https://www.sciencedirect.com/science/article/pii/S2590005623000024>.
- [17] E. Trigili, S. Crea, M. Moisè et al., «Design and Experimental Characterization of a ShoulderElbow Exoskeleton With Compliant Joints for Post-Stroke Rehabilitation», *IEEE/ASME Transactions on Mechatronics*, vol. 24, n.o 4, págs. 1485-1496, 2019. DOI: 10.1109/TMECH.2019.2907465.

- [18] Y. A. Jarrah, M. G. Asogbon, O. W. Samuel et al., «High-density surface EMG signal quality enhancement via optimized filtering technique for amputees' motion intent characterization towards intuitive prostheses control», *Biomedical Signal Processing and Control*, vol. 74, pág. 103 497, 2022, ISSN: 1746-8094. DOI: <https://doi.org/10.1016/j.bspc.2022.103497>. dirección: <https://www.sciencedirect.com/science/article/pii/S1746809422000192>.
- [19] R. Dubey, M. Kumar, A. Upadhyay y R. B. Pachori, «Automated diagnosis of muscle diseases from EMG signals using empirical mode decomposition based method», *Biomedical Signal Processing and Control*, vol. 71, pág. 103 098, 2022, ISSN: 1746-8094. DOI: <https://doi.org/10.1016/j.bspc.2021.103098>. dirección: <https://www.sciencedirect.com/science/article/pii/S1746809421006959>.
- [20] A. Leone, G. Rescio, A. Manni, P. Siciliano y A. Caroppo, «Comparative Analysis of Supervised Classifiers for the Evaluation of Sarcopenia Using a sEMG-Based Platform», *Sensors*, vol. 22, n.o 7, 2022, ISSN: 1424-8220. DOI: 10 . 3390 / s22072721. dirección: <https://www.mdpi.com/1424-8220/22/7/2721>.
- [21] O. Volk, A. Ratnovsky, S. Naftali y G. Singer, «Classification of tracheal stenosis with asymmetric misclassification errors from EMG signals using an adaptive cost-sensitive learning method», *Biomedical Signal Processing and Control*, vol. 85, pág. 104 962, 2023, ISSN: 1746-8094. DOI: <https://doi.org/10.1016/j.bspc.2023.104962>. dirección: <https://www.sciencedirect.com/science/article/pii/S1746809423003956>.
- [22] Álvaro Ángel Orozco Gutiérrez, G. A. Betancourt Orozco, y E. Giraldo Suárez, «Determinación de movimientos a partir de señales electromiográficas utilizando máquinas de soporte vectorial», *Rev. Médica Risaralda*, vol. 11, n.º 2, feb. 2005.
- [23] J. Beunza, E. Sanz y E. Condés, *Manual Práctico de Inteligencia Artificial en Entornos Sanitarios*, 2ª ed. Elsevier Espanol, 2023.
- [24] A. Garcia, “Estudio experimental de diversos algoritmos de aprendizaje por refuerzo”, Universidad de Málaga, 2020.
- [25] I. C. PÃ©rez Verona and L. Arco GarcÃa, ‘Una revisiÃ³n sobre aprendizaje no supervisado de mÃ©tricas de distancia’, *Revista Cubana de Ciencias InformÃ¡ticas*, vol. 10, pp. 43–67, 12 2016.
- [26] Andreas C. MÃ¼ller & Sarah Guido. (2016). Introduction to Machine Learning with Python: A Guide for Data Scientists (1st ed.). O'Reilly
- [27] H. Sahour, V. Gholami, J. Torkman, M. Vazifedan, and S. Saeedi, ‘Random forest and extreme gradient boosting algorithms for streamflow modeling using vessel features and tree-rings’, *Environmental Earth Sciences*, vol. 80, 11 2021.

[28] Y. Huang and L. Li, "Naive Bayes classification algorithm based on small sample set," 2011 IEEE International Conference on Cloud Computing and Intelligence Systems, Beijing, China, 2011, pp. 34-39, doi: 10.1109/CCIS.2011.6045027.

keywords: {Text categorization; Classification algorithms; Bayesian methods; Training; Accuracy; Time frequency analysis; Naïve Bayes; Text classification; Poisson distribution; Classification accuracy; small sample set},

[29] E. M. Kumar Reddy, A. Gurralla, V. B. Hasitha y K. V. Rajesh Kumar, Introduction to Naive Bayes and a Review on Its Subtypes with Applications. Chapman Hall/CRC.

[30] S. V. M. Vishwanathan and M. Narasimha Murty, "SSVM: a simple SVM algorithm," Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No.02CH37290), Honolulu, HI, USA, 2002, pp. 2393-2398 vol.3, doi: 10.1109/IJCNN.2002.1007516.

keywords: {Support vector machines; Iterative algorithms; Kernel; Convergence; Support vector machine classification; Automation; Memory management; Machine learning algorithms; Machine learning; Symmetric matrices}

[31] G. Liu, S. Mao, and J. Kim, 'A Mature-Tomato Detection Algorithm Using Machine Learning and Color Analysis', Sensors, vol. 19, p. 2023, 04 2019.

[32] B. Parhami, "Voting algorithms," in IEEE Transactions on Reliability, vol. 43, no. 4, pp. 617-629, Dec. 1994, doi: 10.1109/24.370218.

keywords: {Voting; Hardware; Algorithm design and analysis; Software algorithms; Redundancy; Distributed computing; Frequency; Computational complexity; Combinatorial mathematics; Fault tolerant systems},

[33] Mohammed Abdul Hafeez Khan, Rohan Varma Rudraraju, Swarnalatha Rajaguru, November 16, 2022, "Bicep Detection Using Electromyography and 3-Axis Accelerometer Data", IEEE Dataport, doi: <https://dx.doi.org/10.21227/7n0e-jm71>.

[34] A. P. Singh and S. Chaudhari, 'Embedded machine learning-based data reduction in application-specific constrained IoT networks', in Proceedings of the 35th Annual ACM Symposium on Applied Computing, Brno, Czech Republic, 2020, pp. 747–753.

[35] Trabajo del Bíceps braquial y su relación con el dolor de espalda - iespalda. (s.f.). iespalda. <https://iespalda.com/trabajo-del-biceps-braquial-y-su-relacion-con-el-dolor-de-espalda/>

[36] Advancer Technologies, "3-lead Muscle / Electromyography Sensor for Microcontroller Applications", MyoWare™ Muscle Sensor (AT-04-001) Datasheet, 2015.

[37] "LM350 Datasheet PDF Download - ON Semiconductor". DatasheetQ.com - Electronic Parts and Semiconductors Datasheet PDF Search Site. Accedido el 1 de junio de 2024. [En línea]. Disponible: <https://www.datasheetq.com/datasheet-download/443474/1/ON-Semiconductor/LM350>