

Aprendizaje de la cinemática de un brazo robótico para la simulación del movimiento mediante
técnicas de Machine Learning

Juan Angel Gualguan Oviedo, ✉ jagualguan.7519@unicesmag.edu.co
Jhonatan Esteban Villareal López, ✉ jhonatan.v.lopez.357@gmail.com

Universidad CESMAG
Facultad de Ingeniería
Ingeniería de Sistemas
Pasto - Colombia
2023

Aprendizaje de la cinemática de un brazo robótico para la simulación del movimiento mediante técnicas de Machine Learning

Juan Angel Gualguan Oviedo, ✉ jagualguan.7519@unicesmag.edu.co
Jhonatan Esteban Villareal López, ✉ jhonatan.v.lopez.357@gmail.com

Informe final como requisito para optar el título de Ingeniero de Sistemas

Asesor:

Mg. Héctor Andrés Mora.

Universidad CESMAG
Facultad de Ingeniería
Ingeniería de Sistemas
Pasto - Colombia
2023

TABLA DE CONTENIDO

INTRODUCCIÓN	13
1. PROBLEMA DE INVESTIGACIÓN.....	14
1.1. Objeto o tema de investigación.	14
1.2. Planteamiento del problema.	14
1.3. Formulación del problema.....	15
1.4. Objetivos.....	15
1.4.1. General	15
1.4.2. Específicos	15
1.5. Justificación	15
1.6. Delimitación	16
2. MARCO TEÓRICO.....	16
2.1. Antecedentes	16
2.1.1. Internacionales	17
2.1.2. Nacionales.....	19
2.1.3. Regionales.....	20
2.2. Supuestos teóricos de investigación Machine Learning (ML).....	21
2.2.1. Robótica	24
2.2.2. Simulación.....	26
2.2.3. Cinemática.....	27
2.3. Variables de estudio.....	27
2.3.1. Cinemática del robot.....	27
2.3.2. Métricas.....	28
2.4. Variable dependiente	28

2.4.1.	Cinemática inversa.....	28
2.5.	Variable independiente.....	29
2.5.1.	Cinemática directa	29
2.6.	Definir normas y variables	29
2.6.1.	Variable dependiente	29
2.6.2.	Variable independiente	30
2.7.	Definición operativa de variables.....	31
2.8.	Formulación de hipótesis.....	32
2.8.1.	Hipótesis investigativa.....	32
2.8.2.	Hipótesis nula.....	32
2.8.3.	Hipótesis alternativa.....	32
3.	METODOLOGÍA	32
3.1.	Paradigma	32
3.2.	Enfoque.....	33
3.3.	Método científico.....	33
3.4.	Tipo de investigación.....	33
3.5.	Diseño investigación.....	33
3.6.	Población	33
3.7.	Muestra	34
3.8.	Técnicas de recolección de información	34
3.9.	Validez.....	34
3.11.	Instrumentos de recolección de información	35
4.	RESULTADOS DE LA INVESTIGACIÓN	35
4.1.	Plan de acción.....	35
4.2.	Plan de acción detallado	38

4.2.1. Objetivo 1	39
4.2.2. Objetivo 2	45
4.2.3. Objetivo 3	47
5. ANÁLISIS Y DISCUSIÓN DE RESULTADOS	52
5.1. Análisis	52
5.2. Discusión	57
6. CONCLUSIONES	58
Bibliografía.....	60
Anexos.....	66

LISTA DE TABLAS

Tabla 1. Datos Tabulares. Matriz de datos.	22
Tabla 2. Plan de acción brazo robótico.	35
Tabla 3. Estructura Dataset tabulares.	45
Tabla 4. Dataset Pruebas para series temporales Multivariantes.	46
Tabla 5. Entrenamiento Red Neuronal con Videos.	56
Tabla 6. Resultados del conjunto de validación.	57
Tabla 7. Comparación función de Perdida para Datasets.	57

LISTA DE FIGURAS

Figura 1. Ramas de la Inteligencia artificial.	21
Figura 2. Datos no tabulares. Imagen de la cinemática del Brazo Robótico de 3 GDL.	23
Figura 3. Cuadro Comparativo Aprendizaje Supervisado y no Supervisado.	24
Figura 4. Ejemplo de Brazo Robótico de 3GDL.	25
Figura 5. Ejemplo de Brazo Robótico con sus articulaciones.	25
Figura 6. Ejemplo de Simulación Brazo Robótico.	26
Figura 7. Diagrama de la investigación.	38
Figura 8. Diagrama de Flujo Simulación Brazo robótico.	41
Figura 9. Selección de letra.	44
Figura 11. Video letra "A".	44
Figura 11. Carga de DATAFRAME.	47
Figura 12. Partición Dataset.	47
Figura 13. Patrones series temporales multivariantes.	49
Figura 14. Pruebas Dataset imágenes con épocas.	52
Figura 16. Resultados con modelo ANN.	53
Figura 17. Resultados con modelo SVM.	54
Figura 18. Resultados con Modelo DT.	54
Figura 19. Resultados con modelo RF.	55
Figura 20. Función de pérdida Dataset de imágenes.	57

Aprendizaje de la cinemática de un brazo robótico para la simulación del movimiento mediante técnicas de Machine Learning

NOTA DE ACEPTACIÓN

DEDICATORIA

Dedico este trabajo de grado a quienes han aportado un grano de arena para que sea posible el cumplimiento del mismo, cada profesor y compañero en las aulas de la Universidad CESMAG, a nuestro asesor y profesor Mg. Héctor Mora, a la facultad de Ingeniería de Sistemas, mientras estuve en la Universidad pude ver como se acopló a los cambiantes tiempos y proveyó nuevo conocimiento para los estudiantes. A mi familia, el apoyo de ellos siempre ha sido incondicional. Este trabajo de grado pretende generar conocimiento y un aporte positivo a la comunidad científica que pueda usar lo desarrollado en esta investigación.

JUAN ANGEL GUALGUAN OVIEDO.

Dedico este trabajo a mis padres, quienes siempre han sido mi fuente inagotable de apoyo y motivación. A mi familia y amigos, por su paciencia y aliento constante a lo largo de esta travesía académica. A mis profesores, por su guía y sabiduría que han iluminado mi camino hacia el conocimiento. A cada desafío superado y cada obstáculo que me ha hecho más fuerte. Este logro es también para todos aquellos que creyeron en mí cuando dudaba de mis propias capacidades. Gracias a cada uno de ustedes, este trabajo es más que una tesis; es el fruto de un esfuerzo colectivo que celebro con gratitud y humildad.

JHONATAN ESTEBAN VILLAREAL LOPEZ.

AGRADECIMIENTOS

Agradezco a Dios por permitirnos culminar este proyecto satisfactoriamente y permitir que sea de este modo la culminación de este caminar por la formación universitaria, Él proveyó todo en todo momento. A mis padres, que me han dado su apoyo y han sido quienes me motivaron a continuar cada momento. A mi compañero de tesis Jhonatan Esteban Villareal López, su amistad y apoyo han sido de gran ayuda. A todos quienes estuvieron en mi recorrido universitario, nuestro asesor Mg. Héctor Andrés Mora Paz quien nos motivó desde semilleros y aportó valiosa información y apoyo en este proyecto.

JUAN ANGEL GUALGUAN OVIEDO.

A mis padres, cuyo amor y apoyo incondicional han sido mi mayor fortaleza a lo largo de este viaje académico. A mis amigos, por compartir risas y aliento durante los momentos difíciles. A mis profesores, por su guía experta y constante inspiración. A todos aquellos que, de alguna manera, contribuyeron a mi crecimiento personal y académico. Este logro es tanto suyo como mío. Agradezco a cada desafío superado, pues me han fortalecido. Esta tesis es un testimonio de la perseverancia, el aprendizaje y la colaboración. Con gratitud, dedico este trabajo a quienes han sido parte integral de mi trayectoria académica.

JHONATAN ESTEBAN VILLAREAL LOPEZ.

RESUMEN ANALITICO DE ESTUDIO (RAE)

Tipo de documento	Trabajo de grado
Tipo de impresión	Digital
Título del documento	Aprendizaje de la cinemática de un brazo robótico para la simulación del movimiento mediante técnicas de Machine Learning.
Autor (es)	Juan Angel Gualguan Oviedo Jhonatan Esteban Villareal López
Asesor	Mg. Héctor Andrés Mora Paz
Lugar donde fue realizado	Universidad CEMAG Facultad de Ingeniería Programa de Sistemas
Palabras clave	Machine Learning, Deep Learning, Brazo robótico, Cinemática, Simulación.
<p>Resumen</p> <p>La investigación tuvo como objetivo desarrollar un simulador del movimiento de la cinemática de un brazo robótico de 3 grados de libertad (3GDL) aplicando técnicas de Machine Learning; con el fin de determinar cuál es la mejor configuración, se realizaron pruebas en dos conjuntos de datos, estos se clasificaron en tabulares, es decir, tablas que contienen las coordenadas del movimiento del brazo robótico de 3 GDL y videos e imágenes (frames) del movimiento del brazo robótico generado por simulación.</p> <p>Los datos tabulares contienen las coordenadas en las que el brazo robótico genero la cinemática realizando trazos en el espacio, formando letras del abecedario (A-Z) 26 letras en mayúsculas, omitiendo la letra Ñ, de los cuales se generaron tablas que fueron acopladas para formar un solo conjunto (datos tabulares), con el cual se probó la eficiencia del tratamiento de los datos por medio de algunos algoritmos de Machine Learning, como lo son Soporte de Vectores (SVM), Redes Neuronales Artificiales (ANN), Arboles de Decision (DT), y Bosques Aleatorios (RF), con los cuales, se puedo determinar la eficiencia del aprendizaje mediante la función de perdida.</p> <p>Los datos no tabulares contienen los videos e imágenes de la simulación del brazo robótico, este conjunto de datos esta conformado por 26 videos, cuyas características fueron extraídas (ancho alto y canales) y se realiza el preprocesamiento donde se divide el conjunto de datos en conjuntos de entrenamiento (80%) y validación (20%), una vez se procesan estos datos mediante un modelo de red Neuronal Convolutional LSTM utilizando keras, y compilado, por último aplica la función de perdida, con el fin de determinar qué tan bien está funcionando el modelo en el conjunto de validación, un valor cercano a 0 indica un mejor rendimiento.</p> <p>Se finalizó realizando una comparación entre los datos generados por la función de perdida para ambos conjuntos, identificando así que el mejor modelo fue entrenado con el dataset de videos e imágenes.</p>	

INTRODUCCIÓN

El presente proyecto pretende realizar una simulación de la cinemática del brazo robot de 3 grados de libertad mediante técnicas de Machine Learning aplicadas a conjunto de datos tabulares y videos e imágenes. El movimiento de las articulaciones de un robot para desarrollar una determinada tarea es calculado a través de un conjunto de coordenadas dadas a partir de un sistema de referencia ya sea en coordenadas cartesianas, cilíndricas o esféricas y a partir de allí se calculan la cantidad de desplazamientos de cada articulación del robot, a lo que se le conoce como cinemática inversa.

Si se desea que el robot aprenda a realizar estos cálculos por sí solo es necesario optar por otras alternativas como: Aproximaciones de movimiento cortas con retroalimentación sensorial y aprendizaje automático. En esta segunda opción es donde se enfoca el presente estudio, no obstante, para ello es necesario contar con un conjunto de datos que permita realizar el entrenamiento que proporciona a un robot de la capacidad de aprender. Como un acercamiento a tareas más complejas y ante la escasez de un conjunto de datos para la escritura de brazos robóticos de 3 grados de libertad, en esta investigación se realizó una comparación entre algoritmos de aprendizaje automático capaces de resolver una tarea de cinemática inversa y evaluar el grado de exactitud y tiempos de inferencias.

1. PROBLEMA DE INVESTIGACIÓN

1.1. *Objeto o tema de investigación.*

Aprendizaje automático de la cinemática directa e inversa de un brazo robótico.

1.2. *Planteamiento del problema.*

El movimiento de las articulaciones de un robot para desarrollar una determinada tarea es calculado a través un conjunto de coordenadas (datos) dadas a partir de un sistema de referencia ya sea en coordenadas cartesianas, cilíndricas o esféricas y a partir de allí se calculan las cantidades de desplazamiento de cada articulación del robot, a lo que se le conoce como cinemática inversa [1]. Estos cálculos generalmente son realizados por agentes externos al robot ya sea un humano o software [1], esto requiere que los datos ingresados o capturados sean precisos y para ello es necesario aplicar protocolos y técnicas que permitan alcanzar la exactitud deseada, de lo contrario se generan configuraciones erróneas. Otra alternativa para obtener la cinemática de un robot es utilizando técnicas de aprendizaje automático, en esas técnicas se requiere conjuntos de datos que permitan entrenar modelos con un error bajo. Sin embargo, al buscar conjuntos de datos para el entrenamiento de robots escritores de tres grados de libertad, las cadenas de búsqueda arrojaron conjuntos de datos para otros tipos de tareas y con robots de otras características mecánicas[2][3]. Adicional a lo anterior, generalmente para que un robot solucione una tarea específica ha sido previamente programado para realizar los movimientos adecuados para resolverla, esto conlleva a que por cada tarea nueva que exista se necesita la intervención de un agente externo que realice la reconfiguración y por lo tanto no se produce aprendizaje automático. Sin embargo, hay esfuerzos por suplir la intervención de un agente externo utilizando técnicas de Machine Learning, pero se desconoce si es mejor utilizar datos matriciales (estructurados) o videos (no estructurador) para el entrenamiento, como también cuál algoritmo tiene el menor compromiso entre el error y tiempo de inferencia.

1.3. *Formulación del problema.*

¿Cuál es la mejor configuración para el aprendizaje automático de la cinemática inversa de un robot de 3 grados de libertad?

1.4. *Objetivos*

1.4.1. *General*

Determinar si la mejor configuración de aprendizaje de la cinemática directa e inversa de un brazo robótico simulado de 3 GDL, para trazos escritos en el espacio, es ofrecida por entrenamientos realizados en video mediante la comparación de algoritmos de Machine Learning configurados en conjuntos de datos de frames (video) o tabulares.

1.4.2. *Específicos*

- Construir un simulador de un robot de 3 grados de Libertad (GDL) que resuelva un conjunto de trazos finos en el espacio utilizando el algoritmo de Denavit - Hartenberg.
- Generar datasets del movimiento de trazos finos en video o coordenadas cartesianas para el entrenamiento de la cinemática inversa de un robot de 3 GDL adquiridas mediante un robot simulado.
- Comparar el compromiso entre el error y el tiempo de inferencia de un conjunto de algoritmos de Machine Learning recomendados por expertos para la selección de las mejores configuraciones en los conjuntos de datos construidos.

1.5. *Justificación*

Para que el presente estudio tenga los resultados previstos es de gran ayuda la construcción de un simulador de un robot de tres grados de libertad (3 GDL), el cual proporcionará los resultados en lo que se denomina cinemática inversa haciendo énfasis en los trazos finos en el espacio utilizando el algoritmo de Denavit-Hartenberg. Además de brindar una solución sostenible al estudio este simulador genera un aporte al área de Machine Learning.

Para la obtención de resultados en el simulador se debe generar un dataset del movimiento de trazos finos en video o coordenadas de la cinemática inversa de un robot escritor de 3 GDL, lo cual cubre la pobre cantidad de datos con respecto a la cinemática inversa de un robot escritor de 3 GDL, evitando así la tediosa tarea de generar un dataset diferente para cada modelo.

Una vez obtenido y analizado el dataset se debe comparar la relación error y el tiempo de inferencia de un conjunto de algoritmos de Machine Learning; estos algoritmos son previamente evaluados y seleccionados por expertos en el área, para así aplicar las mejores configuraciones en los conjuntos de datos construidos. configuraciones que pueden ser usadas en diferentes modelos con un respectivo estudio que respalde su uso.

1.6. Delimitación

Este proyecto se realizará por medio del servicio de Google, Google Collaboratory, el cual es un servicio gratuito alojado en la nube. Por medio de este se genera el Script (código fuente) y se llevarán a cabo las pruebas experimentales para dar cumplimiento a los objetivos propuestos, además de resolver las hipótesis planteadas.

Para este proyecto el tiempo aproximado de duración es de 10 meses de febrero a noviembre de 2023.

2. MARCO TEÓRICO

2.1. Antecedentes

En ciencias de la computación existe un proceso al cual se ha denominado Inteligencia Artificial (IA), haciendo referencia a los procesos que desarrollan la inteligencia, pronunciado oficialmente por el informático John McCharty en 1956. Andreas Kaplan y Michael Haenlein definen la inteligencia artificial como la capacidad de un sistema para interpretar correctamente datos externos, para aprender de dichos datos y emplear esos conocimientos para lograr tareas y metas concretas a través de la adaptación flexible [4].

En la actualidad muchas empresas de gran magnitud como Amazon con su producto AWS, la cual presta servicios de desarrollo y soporte en la nube [5], Microsoft con su producto AZURE la cual provee soluciones en la nube y en competencia con el anterior [6] y empresas afines al desarrollo tecnológico prestan servicios que involucran IA. Una de las ramas de la IA es el Machine Learning (ML) (Aprendizaje Automático), según Russell y Norving definen al Machine Learning como un computador observa datos, construye un modelo basado en esos datos y utiliza ese modelo a la vez como una hipótesis acerca del mundo y una pieza de software que puede resolver problemas [7]. Ya que el proyecto se llevará a cabo mediante técnicas de ML, también es importante mencionar el Deep Learning (DL) (Aprendizaje Profundo), el cual es un conjunto de algoritmos del ML, según documentación el DL se define como un conjunto de cálculos de adiestramiento autónomo (Machine Learning) que intenta modelar apercpciones de alto nivel en apuntes usando arquitecturas computacionales que admiten alteraciones no lineales diferentes e iterativas de documentos expresados en apariencia matricial o tensorial [8], el aprendizaje profundo (DL) es el subcombustible del aprendizaje automático (ML) que se centra en modelos de redes neuronales profundas [9], es decir, los datos que se van a estudiar y la arquitectura de algoritmos de Deep Learning son una combinación muy útil en la investigación para arrojar datos más precisos a medida que se genera la experimentación e implementación en la simulación del brazo robótico.

2.1.1. Internacionales

A lo largo del mundo se llevan a cabo múltiples estudios con los temas relacionados a la robótica y el Machine Learning, es posible observar en los siguientes proyectos, uno de ellos titulado **“Diseño y fabricación de robot didáctico programable de 3 grados de libertad”**, donde Hernández, Becerra, Dublán, Mata, Morales, Franco mencionan que el proyecto usa diferentes instancias, las cuales les ayudan a lograr el objetivo que es el lograr una precisión o exactitud en cuanto al brazo robótico [10], para el presente proyecto permite

observar cuáles fueron los datos utilizados en las coordenadas para lograr que el brazo robótico tenga un error mínimo en el movimiento.

Así mismo otro proyecto titulado “**Diseño, montaje, implementación y simulación de un brazo robótico con 3 grados de libertad con fines educativos.**”, en el cual Santamaría, menciona que, en el funcionamiento del robot, hay diversas funciones las cuales ayudan a que este puede moverse y desempeñarse en cada movimiento, una de estas funciones es sysCall Actuation; el cual hace que la precisión del robot sea bastante alta [11] y esto para el proyecto actual puede proporcionar funciones que pueden ser utilizadas para realizar la simulación del brazo robótico de 3 GDL.

Otro proyecto del cual se toma referencia titula “**Diseño e implementación de las funciones de agarre y levante en un brazo kinova usando señales EEG y Deep Learning**”, para el diseño de levantamiento del bazo se debe hacer uso del algoritmo más eficiente, para ello se hará una comparación entre los algoritmos más evolucionados para cumplir esta función y se escogerá el mejor, una de las comparaciones que se han realizado tiempo atrás es el que hizo Neyra, en su investigación se comparan tres algoritmos también entrenados con el dataset denominado WAYEEG-GAL, uno de los cuales está basado en una red neuronal convolucional, la cual utilizaremos como benchmark. Este modelo tomó 60 minutos en clasificar 3 '144000 muestras, equivalente a 104.8 minutos de adquisición de señales EEG; en otras palabras, clasifica 52400 muestras/min. Asimismo, el desempeño de este modelo en la métrica AUROC fue de 0.829 [12], esto es de gran ayuda para el proyecto actual, ya que esto sirve de ejemplo para comparar qué algoritmos de aprendizaje automático (ML) y profundo (DL) fueron usados y cómo se comportó el brazo robótico con esos datos.

El proyecto que se llevará a cabo no es el primero en realizarse, pero cabe mencionar que, basándose en la documentación, busca aportar un conjunto de datos que faciliten a los investigadores realizar más pruebas en la automatización de procesos mediante técnicas de Machine Learning.

2.1.2. Nacionales

Con lo anterior mencionado, cabe recalcar que los investigadores a nivel nacional buscan aportar nuevas formas de aplicar el tratamiento a los conjuntos de datos con el fin de que estos sean aprovechados por las máquinas y quienes las manipulan para realizar tareas, por ejemplo, el artículo titulado **“Desarrollo matemático y aplicación de la cinemática inversa por medio de un software de cálculo y programación, para el robot Mitsubishi rv-m1”**, Gómez y Bustos, plantean la solución de la cinemática inversa y directa de un manipulador Melfa Mitsubishi RV-M1, a través de una interfaz de operación. Se comienza con una recopilación de la investigación realizada del robot donde se incluye la solución de la cinemática y la explicación de la manipulación por medio del Software, de esto se toman datos guías. De esta forma se realiza la solución por medio de un programa de cálculo que permite al usuario observar en una interfaz gráfica la ubicación inicial y final del manipulador [13]. Este artículo es de gran utilidad ya que se plantea una solución manipulando datos ya previamente obtenidos del movimiento del robot y genera un modelo predictivo, esta misma metodología puede ser trasladada al presente proyecto.

El proyecto también trata temas como la cinemática inversa del robot, de la cual se debe obtener un modelo para generar una simulación del movimiento del Brazo Robótico, un referente se encuentra en el proyecto titulado **“UR3 modelo cinemático inverso”**, Romero, Páez y Granizo en el cual mencionan que se presenta la investigación y el cálculo del modelo cinemático inverso del manipulador industrial UR3 mediante un método geométrico, el cual posteriormente realizará una simulación mediante una comunicación entre Python y CoppeliaSim [14]. Para el presente proyecto aporta conocimientos esenciales de simulación y cinemática inversa.

Así se observa que es necesario acudir a las máquinas como medios de ayuda que aporten en la industria y también en la educación como lo mencionan en el proyecto titulado **“DIDAPLAT: plataforma robótica de bajo costo con 3 GDL para fines educativos.”**, Navarro y Avalos en el cual, presentan la creación de un prototipo de robot paralelo de 3 GDL que pueda ser empleado como medio de enseñanza, con un enfoque de bajo costo y

la posibilidad de seguimiento de trayectorias [15]. Este proyecto tuvo en cuenta las ecuaciones de cinemática inversa y su obtención para ser ejecutado correctamente, temas que son necesarios en el presente proyecto.

2.1.3. Regionales

El proyecto en proceso busca la automatización completa de la máquina; mediante diferentes técnicas de Machine Learning se puede lograr que un Robot recree la naturaleza humana de aprender por sí solo. En la ciudad de San Juan de Pasto y a lo largo del departamento se observa que la necesidad de automatizar procesos industriales, donde las máquinas suelen ser manipuladas por operarios expertos, ha sido impactada también por procesos realizados con IA y Machine Learning.

El proyecto de grado realizado por estudiantes de la Corporación Universitaria Autónoma de Nariño, el cual titulan **“Diseño y construcción de un prototipo de robot paralelo orientado a procesos siderúrgicos de corte de lámina, a través del acondicionamiento de equipo de oxicorte, en la ciudad de san juan de pasto”**, Belalcázar, Botina y Zambrano donde mencionan que el prototipo disminuirá considerablemente el contacto directo hombre-máquina, por ser un dispositivo semiautomático que en conjunto con sistemas electromecánicos, electrónicos e informáticos es capaz de ejecutar acciones en la lámina sin manipulación directa del encargado, además de garantizar la calidad en el producto final [16]. Para el presente proyecto este referente local permite observar qué alcances tuvo el desarrollo y en qué áreas se ha buscado la automatización de las máquinas. Otro de los referentes es un artículo, de la Universidad de Nariño, titulado **“Uso de sensores inerciales en fisioterapia: una aproximación a procesos de evaluación del movimiento humano”**, Castellanos, Montealegre, Martínez, Gallo y Almanza mencionan que el objetivo fue la revisión de literatura para analizar la información sobre el uso de sensores inerciales como una aproximación a los procesos de evaluación del movimiento corporal humano [17]. Esto es de utilidad en el presente proyecto, para hacer un análisis de los datos y reconocer cuando hay fallas en el resultado de una predicción.

A pesar de que en la región se ha incrementado el desarrollo e implementación de nuevas tecnologías, no se evidencian muchos proyectos relacionados con el tema de esta investigación, o por lo menos no se han publicado en los repositorios de las universidades locales.

2.2. *Supuestos teóricos de investigación Machine Learning (ML)*

El Machine Learning (ML) hace referencia a una disciplina derivada del campo de la Inteligencia Artificial (IA) el cual, a través de distintos algoritmos, brinda la capacidad de identificar patrones en datos masivos y elaborar predicciones, generalmente usado en la robótica o modelos predictivos en ordenadores, el Machine Learning es un campo vital de la inteligencia artificial.

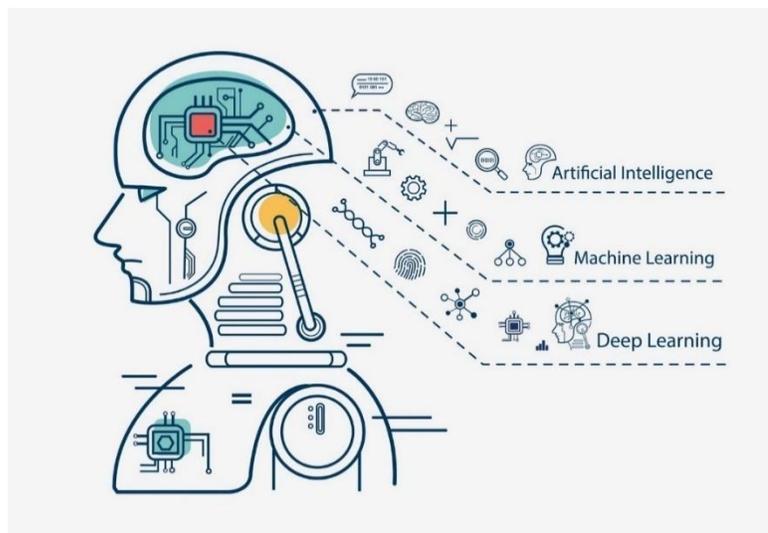


Figura 1. Ramas de la Inteligencia artificial.

Fuente. Extraído de <https://www.atrainingnovation.com/wp-content/uploads/2021/02/portada-1080x675.jpg.webp>

El presente estudio tratará los temas como el aprendizaje supervisado el cual será de bastante ayuda para que el robot tenga un aprendizaje mediante distintas métricas, según Nguyen [18], la finalidad del aprendizaje supervisado de funciones de distancia es aprender una función de distancia, por lo general la distancia de Mahalanobis, entre dos instancias y sus clases, aplicándolas de manera específica, usando para ello información del conjunto de entrenamiento. Para esto la mayoría de los algoritmos que aprenden una función de distancia tratan de resolver un problema de optimización con restricciones, haciendo uso de datos tabulares como videos o imágenes, al igual que datos no tabulares como matrices y grafos.

Dentro del Aprendizaje Supervisado, se encuentran los datos tabulares y no tabulares, según documentación los datos tabulares pueden ser aquellos que se registran en tablas de verdad o meramente registros en una tabla la cual contiene unas marcas específicas que denotan ausencia o presencia del dato, estas marcas pueden ser categóricas (Cliente, Producto, Marca, Edad, Género) o numérica (1,2,3...). [19].

Tabla 1. Datos Tabulares. Matriz de datos.

Datos Tabulares					
	Categórica				
	Cliente	Edad	Género	Producto	Marca
Numérica	1	23	M	Pan	Artesanal
	2	23	F	Soya	Launus
	3	22	F	Camisa	Reg
	4	18	M	Juguete	Toys
	5	34	M	Ninguno	Ninguno

Fuente. Investigación Propia.

También se encuentran los datos No tabulares que en esta investigación serán aquellos datos representados por imágenes y videos, a continuación, puede observar un ejemplo de la simulación del brazo robótico de dos grados de libertad.

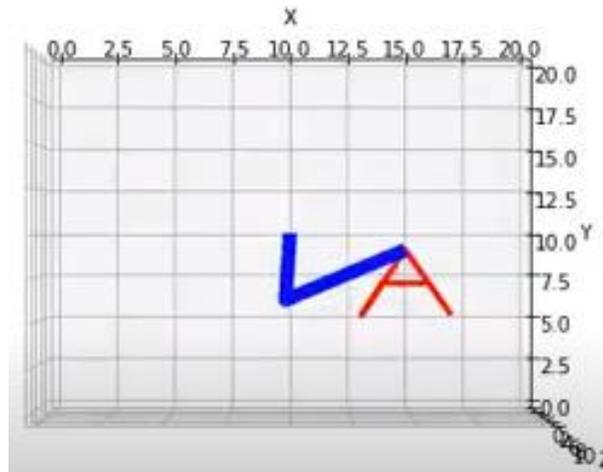


Figura 2. Datos no tabulares. Imagen de la cinemática del Brazo Robótico de 3 GDL.

Fuente. Investigación Propia, Simulación brazo robótico

Además de hacer uso de aprendizaje supervisado también se hará uso de aprendizaje no supervisado, ya que la simulación del brazo robótico también se aplicarán algoritmos de este tipo, por medio de Deep Learning o visión artificial; esta forma de aprendizaje tiene bastantes ventajas, como dijo Gómez [20], Lo que se desea, es hallar ciertas estructuras que sean capaces de caracterizarlos. Este tipo de aprendizaje es muy utilizado en los problemas de clustering (agrupación de ítems con características similares), en el cual se deben agrupar los datos, muchas veces dichas agrupaciones no tienen un significado real. Es importante notar que, la correlación no implica causalidad y generalmente se pueden dar falsos positivos, es decir, hallar estructuras entre datos que no infieran ninguna relación real. [21][22].

Workflow

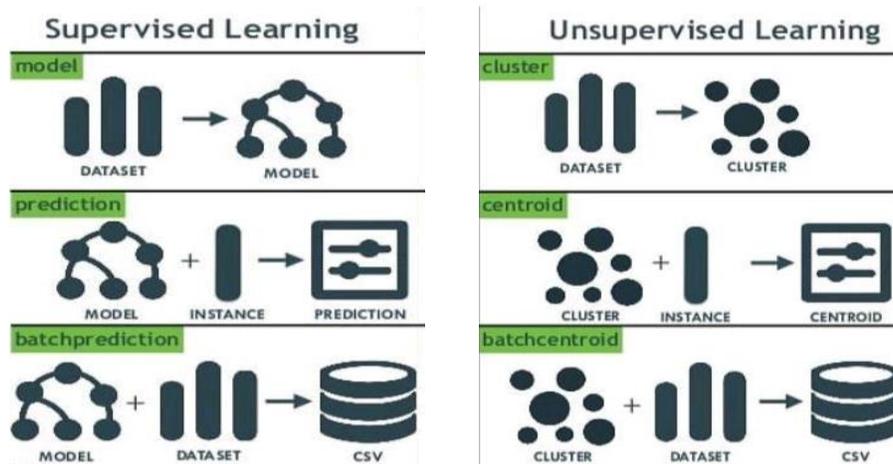


Figura 3. Cuadro Comparativo Aprendizaje Supervisado y no Supervisado.

Fuente. Extraído de <https://dl.asmaamir.com/quick-visual-info>

2.2.1. Robótica

El robot es una Máquina automática programable capaz de realizar determinadas operaciones de manera autónoma y sustituir a los seres humanos en algunas tareas, para el presente estudio se hará uso de un robot de tipo articulado, el cual va a simular un brazo humano, este tendrá 3 grados de libertad de movimiento para ser ejecutado en un plano bidimensional o tridimensional [23].

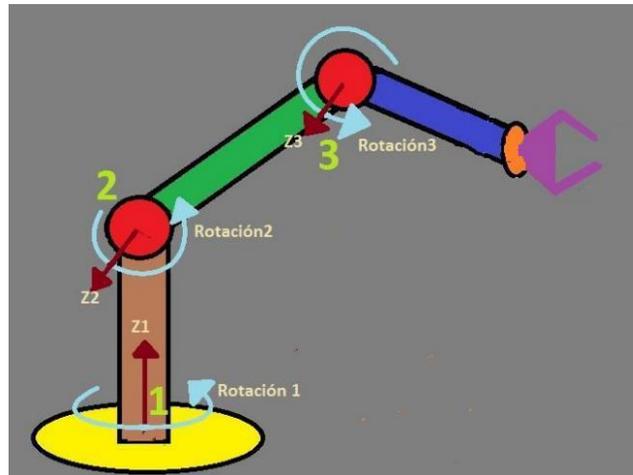


Figura 4. Ejemplo de Brazo Robótico de 3GDL.

Fuente. Extraído de <https://docplayer.es/155249207-Diseno-e-implementacion-de-brazo-robotico-industrial-arduino.html> (pág. 4).

Se hará un estudio con base a los algoritmos que provocan el movimiento en el robot, Uno de los factores importantes a tener en cuenta es la cinemática del robot, a lo que se refiere es el estudio de los movimientos realizados por el robot, donde se emplean la cinemática inversa y directa, con la primera se realizará este proyecto, puesto que, en ella se encuentra el estudio del movimiento que llevará a cabo el algoritmo, una vez ingresados los datos, con lo cual se posiciona haciendo una análisis con base a la articulaciones del robot. [23].

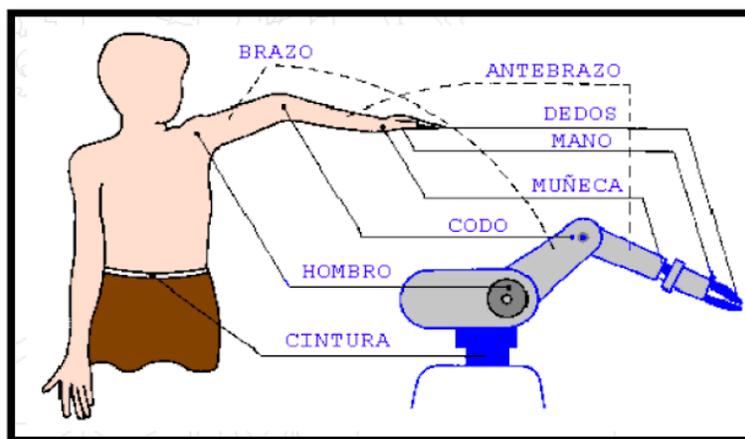


Figura 5. Ejemplo de Brazo Robótico con sus articulaciones.

Fuente. Extraído de http://platea.pntic.mec.es/vgonzale/cyr_0204/cyr_01/robotica/sistema/morfologia.htm

Este estudio se apoya en la prueba y evaluación de cada algoritmo, haciendo uso en técnicas como la que uso [24]:

“En este segmento de código se expone cómo el programa provee las dos posibles soluciones al problema cinemático inverso recibiendo como entrada las especificaciones físicas del robot y las coordenadas del punto al cual se desea llegar. Para fines de evaluación y veracidad se puede hacer uso de la cinemática directa y evaluar la reciprocidad esperada en la cinemática inversa, Aplicando una rotación para valores específicos se obtiene una posición del efecto final.” [24].

2.2.2. Simulación

Las simulaciones pueden utilizarse como experiencias previas al trabajo experimental de laboratorio para darles una idea de lo que encontrarán en un experimento real. Esto contribuye a incrementar la seguridad de los laboratorios, producir ahorros financieros al reducir el tiempo que requieren los estudiantes en el laboratorio y por tanto el número de estaciones de trabajo requeridas [25]. El proyecto tendrá como objetivo usar simulaciones en computador para poder realizar los estudios de los diferentes resultados, tomando en cuenta características propias como la precisión, dirección, fuerza etc., para que así se pueda elegir la mejor configuración algorítmica, la más pertinente a la hora de crear trazos precisos.

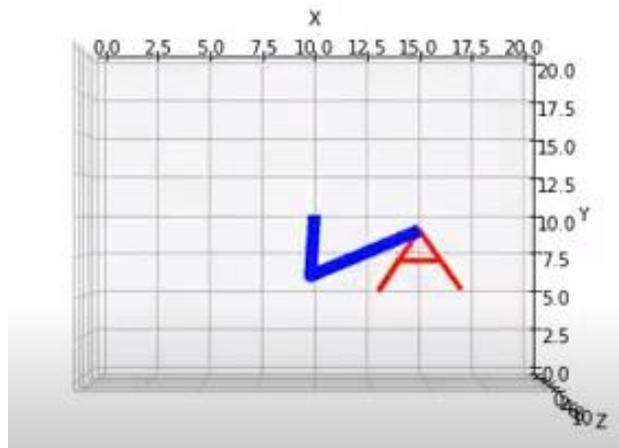


Figura 6. Ejemplo de Simulación Brazo Robótico.

Fuente. Investigación Propia, Simulación brazo robótico. Realizando trazos de letra A

2.2.3. *Cinemática*

La cinemática dentro de la Física estudia el movimiento de los cuerpos, es decir analizando su desplazamiento, aunque sin fijarse en sus dimensiones.

Este movimiento parte de un punto de referencia originado en un sistema de coordenadas, este sistema no es más que dos o tres rectas imaginarias que se cruzan entre sí.

2.3. *Variables de estudio*

En la investigación se medirán las siguientes variables:

2.3.1. *Cinemática del robot*

Cinemática directa: Sirve para calcular las coordenadas de los eslabones a partir de las coordenadas articulares (Ángulos).

X : Variable que representa la posición del extremo final del brazo.

$frame_i$: Cuadro específico (imagen) del video.

x_{i1}, x_{i2}, x_{i3} : Coordenada específica del eslabón del robot.

t : Tiempo de captura del cuadro (Frame) del video.

i : Índice que indica numero de cuadros (frames) del video.

$$X = frame_i (videos) (1)$$

$$X = x_{i1}, x_{i2}, x_{i3}, t = tiempo, i = \{0,1,2\}(matrlz)$$

Cinemática inversa: Sirve para calcular los desplazamientos articulares a partir de las coordenadas de los eslabones.

y : Variable que representa los ángulos de las articulaciones del brazo robótico.

θ_0 : Función que calcula los ángulos de las articulaciones directamente de las coordenadas X y tiempo t .

θ_1 : Función que calcula los ángulos de las articulaciones a partir de las coordenadas X y los ángulos de la articulación previa θ_0 .

θ_2 : Función que calcula los ángulos de las articulaciones a partir de las coordenadas X y los ángulos de la articulación previa θ_1 .

$$y = [\theta_0, \theta_1, \theta_2]$$

donde:

$$\theta_0 = f_1(X, t)$$

$$\theta_1 = f_2(X, f_1)$$

$$\theta_2 = f_3(X, f_2)$$

2.3.2. Métricas

$$\text{Error Cuadrático Medio (MSE)} = \frac{1}{n} \sum (y - \hat{y})^2 \quad (2)$$

$$\text{Coeficiente de determinación (r}^2\text{)} = 1 - \frac{\text{MSE}(\text{modelo})}{\text{MSE}(\text{Línea base})} \quad (3)$$

$$\text{Data Time Warping (DTW)} \quad (4)$$

2.4. Variable dependiente

2.4.1. Cinemática inversa

$$y = [\theta_0, \theta_1, \theta_2]$$

donde:

$$\theta_0 = f_1(X, t)$$

$$\theta_1 = f_2(X, f_1)$$

$$\theta_2 = f_3(X, f_2)$$

$$T_i + 1$$

$$\text{Error Cuadrático Medio (MSE)} \quad (2)$$

$$\text{Coeficiente de determinación (r}^2\text{)} \quad (3)$$

$$\text{Data Time Warping (Dtw)} \quad (4)$$

2.5. *Variable independiente*

2.5.1. *Cinemática directa*

$$X = frame_i (videos)$$

$$X = x_{i1}, x_{i2}, x_{i3}, t = tiempo, i = \{0,1,2\}(matrlz)$$

Algoritmos: SVM, ANN, Decision Tree, Random Forest

2.6. *Definir normas y variables*

2.6.1. *Variable dependiente*

Coordenadas cartesianas (X): En la cinemática de un robot hace referencia a las posiciones de cada una de sus articulaciones en un sistema de referencia [26].

Error cuadrático medio (ECM): Es el promedio de la diferencia de los cuadrados entre una variable real y una variable predicha, se mide como se indica en la ecuación (2) [27].

Para medir los frames y las coordenadas cartesianas se utilizará los cambios de posición entre lapsos de tiempo.

Cinemática inversa: Sirve para calcular los desplazamientos articulares a partir de las coordenadas de los eslabones. Donde θ (*Teta*) son los desplazamientos angulares de cada una de las articulaciones. Se mide a partir de las diferencias angulares entre lapsos de tiempo [20].

Coefficiente de determinación (r^2): Se define como la diferencia entre un modelo perfecto (1) y la relación entre el error cuadrático medio del modelo y el error cuadrático medio de la línea base (promedio entre el cuadrado de la diferencia entre los valores reales y el promedio de los datos) [28].

Data time warping (Dtw): Es un algoritmo con el cual se miden similitudes entre dos secuencias temporales, estas secuencias pueden ser vídeo, audio o gráfico. Es utilizada ampliamente en analítica de datos (Analytics) es una rama del análisis de datos (Data Analysis). Con este algoritmo se puede encontrar el coste mínimo al comparar series de tiempo y así definir el camino más óptimo[29].

2.6.2. *Variable independiente*

Cinemática directa: Sirve para calcular las coordenadas de los eslabones a partir de las coordenadas articulares (Ángulos) [30].

Frame (X): Es un conjunto de imágenes correlacionadas en el tiempo, para medirlas se descomponen los videos en un vector de imágenes ordenados secuencialmente por la marca temporal que corresponde. Para realizar entrenamientos de series temporales se lo toma como el conjunto de características. [24]

SVM: El algoritmo de máquinas de vectores de soporte (SVM, por sus siglas en inglés) es un algoritmo de aprendizaje supervisado utilizado en el campo de la inteligencia artificial para la clasificación y la regresión. SVM es uno de los algoritmos más populares en el aprendizaje automático debido a su capacidad para manejar eficazmente problemas de alta dimensión [31], [32].

ANN: El algoritmo ANN (Artificial Neural Network) es una técnica de aprendizaje supervisado utilizada en inteligencia artificial para la clasificación, regresión y procesamiento de datos. Las redes neuronales artificiales se componen de múltiples capas de neuronas interconectadas, que trabajan juntas para aprender y realizar tareas complejas.

Cada neurona en la red toma una entrada y realiza una operación matemática para producir una salida. Las neuronas están conectadas entre sí mediante pesos sinápticos, que determinan la fuerza y dirección de la señal de entrada. Durante el proceso de entrenamiento, la red neuronal ajusta los pesos de las conexiones para minimizar el error entre la salida deseada y la salida real [33].

Decision Tree: El algoritmo Decision Tree (dt, Árbol de Decisión), es una técnica de aprendizaje supervisado utilizada en la minería de datos y el análisis de datos. El objetivo principal de un árbol de decisión es crear un modelo predictivo que pueda utilizarse para clasificar nuevas instancias.

Un árbol de decisión se construye a partir de un conjunto de datos de entrenamiento, donde cada instancia está representada por un conjunto de atributos y una etiqueta de clase. El árbol se construye dividiendo el conjunto de datos en subconjuntos cada vez más pequeños y homogéneos, de manera que cada subconjunto contenga instancias similares. La división se realiza seleccionando el atributo que mejor discrimine entre las clases de la etiqueta y dividiendo el conjunto de datos según los valores de ese atributo.

El resultado final es un árbol de decisión en el que cada nodo interno representa una prueba en un atributo, cada rama representa el resultado de la prueba y cada hoja representa una etiqueta de clase. Para clasificar una nueva instancia, se recorre el árbol de decisión siguiendo las pruebas en los atributos hasta llegar a una hoja, que proporciona la etiqueta de clase correspondiente. [34], [35].

Random Forest: El Random Forest (rf, bosque aleatorio) es un algoritmo de aprendizaje automático que combina la idea de Bagging (ensacado) con la selección aleatoria de características, lo que lo hace más robusto y generalizable.

El algoritmo crea múltiples árboles de decisión aleatorios a partir del conjunto de datos de entrenamiento, seleccionando de forma aleatoria un subconjunto de características en cada iteración y ajustando cada árbol a un subconjunto diferente de los datos de entrenamiento.

Cuando se realiza una predicción, el Random Forest promedia las predicciones de todos los árboles de decisión para producir una predicción final. El Random Forest es utilizado en una variedad de tareas de clasificación y regresión, y ha demostrado ser efectivo en la identificación de patrones complejos en grandes conjuntos de datos. [36], [37].

2.7. Definición operativa de variables

Las coordenadas cartesianas de un robot simulado, se van a medir tomando los datos de las articulaciones de un robot de 3 GDL implementando un script. Esto será aplicado para la cinemática inversa y directa.

2.8. Formulación de hipótesis

2.8.1. Hipótesis investigativa

Haciendo uso de un brazo robot de 3 grados de libertad simulado se concluye que los movimientos realizados mediante técnicas de Machine Learning con un conjunto de datos como videos e imágenes tiene un margen de error menor o igual al de un robot entrenado con datos tabulares.

2.8.2. Hipótesis nula

Haciendo uso de un brazo robot de 3 grados de libertad simulado se concluye que los movimientos realizados mediante técnicas de Machine Learning con un conjunto de datos como videos e imágenes tiene un margen de error mayor o igual al de un robot entrenado con datos tabulares.

2.8.3. Hipótesis alternativa

Haciendo uso de un brazo robot de 3 grados de libertad simulado se concluye que los movimientos realizados mediante técnicas de Machine Learning con un conjunto de datos como videos e imágenes tiene un margen de error similar al de un robot programado.

3. METODOLOGÍA

3.1. Paradigma

El paradigma que se utiliza en este proyecto es positivista, también llamado paradigma prediccionista, pues, busca mediante hipótesis estudiar y verificar la naturaleza de fenómenos y así demostrar a través del método científico cuan verídicos son los datos que arroja. Para este proyecto se presentaron hipótesis en a los posibles resultados que se pudo concluir [38] [39].

3.2. *Enfoque*

El presente proyecto está basado en el enfoque cuantitativo y señala que, bajo la perspectiva cuantitativa, la recolección de datos es equivalente a medir. Es decir, adquirir un conjunto de datos al que se le van a realizar distintas pruebas en tiempo real. [40][41].

3.3. *Método científico*

La investigación científica se lleva a cabo con el método científico o empírico-analítico, es decir, busca a través de la observación de fenómenos en la realidad, describirlos sin ánimo de lucro emocional o económico. El presente proyecto aplica el método científico por el análisis que se realizará a los datos y la observación al tratamiento de esos datos en el brazo robótico. [42] [43].

3.4. *Tipo de investigación*

Esta investigación es de tipo correlacional, ya que cuenta con una cantidad de variables de estudio tanto dependientes como independientes, se pretende, hacer predicciones teniendo en cuenta los resultados de diferentes pruebas de algoritmos haciendo uso de muestras de datasets. [44][45]

3.5. *Diseño investigación*

Este proyecto tiene un diseño cuasiexperimental, debido a que se realizó una comparación en los resultados que arrojaron las pruebas a los datos tabulares (Matrices) y los no tabulares (Frames) con base a las métricas de MSE (2), r^2 (3) y DTW (4). [39] [40]

3.6. *Población*

Para este proyecto fue necesario implementar el paradigma del Data-Driven Approach, ya que en este caso los datos previamente capturados será el dataset de

la cinemática inversa creada en la simulación del brazo robótico de 3 grados de libertad [46].

3.7. *Muestra*

Como se menciona anteriormente, el paradigma del Data-Driven Approach hace que tanto Población y Muestra no sean necesarios ya que, la investigación se basará en los datos generados por la simulación [46].

3.8. *Técnicas de recolección de información*

En esta investigación se utilizó la técnica de observación, los resultados de las simulaciones serán clasificados, aunque el interés del proyecto es que se haga una selección automática de los mejores conjuntos de datos para realizar una caracterización de estos [47].

3.9. *Validez*

El presente estudio tomó datos de un dataset preseleccionado para la recolección de información, ya que, este se enfoca más en lo que puede lograr con estos datos. Será revisado por el ingeniero Héctor Andrés Mora Paz, quien verificará los datos obtenidos con los algoritmos aplicados en la cinemática del robot usando datos matriciales. Este proyecto es de tipo cuasiexperimental ya que se tiene que indagar en diferentes repositorios y diferentes autores los cuales proporcionan información de vital importancia [48].

3.10. *Confiabledad*

Este proyecto es de tipo cuasi experimental, por lo que mediante técnicas de Machine Learning se arrojaran datos que reduzcan en gran manera un posible porcentaje de error, esto mediante la configuración del algoritmo, además, con base a los datos que arrojaron las pruebas se validaron o negaron las hipótesis planteadas [49].

3.11. *Instrumentos de recolección de información*

La recolección de información para este proyecto serán los datos que proporcione el dataset generado por la simulación del brazo robótico. También es necesario entender cómo se hará la creación de este dataset, para realizar esta tarea es necesario preparar los datos y así tener un conjunto de entrenamiento el cual debe separar cada vídeo obtenido de varios frames y guardarlos en carpetas separadas. Después se hará uso de una red neuronal ya entrenada para procesar estos frames, para detectar la cinemática del brazo robótico [46].

4. RESULTADOS DE LA INVESTIGACIÓN

Este proyecto se ha desarrollado mediante una metodología de investigación científica, con la cual se han realizado los aportes necesarios para generar la información pertinente en este informe o trabajo de grado. Mediante la simulación del movimiento de un brazo robótico que realizó trazos en el espacio; trazos con los que se dio forma a las 26 letras del abecedario, desarrollado con la metodología en cascada. También a lo largo del proyecto se generaron datos, los cuales son la base investigativa para el desarrollo de las pruebas con Machine Learning, por sugerencia del asesor Mg. Héctor Andrés Mora Paz, se ha determinado que para el desarrollo de este proyecto se utilizara la metodología CRISP-DM (Cross Industry Standard Process for Data Mining). A continuación, se menciona el plan de acción con el que se generó el simulador del brazo robótico y el análisis de datos.

4.1. *Plan de acción*

Tabla 2. Plan de acción brazo robótico.

Fase	Actividades	Recursos	Producto
Objetivo 1 Construir un simulador de un robot de 3 grados de Libertad (GDL) que resuelva un	<ol style="list-style-type: none"> 1. Recolección Requisitos funcionales y no funcionales para el simulador. 2. Análisis y diseño del simulador. 	Lenguaje: Python Bibliotecas: Google.colab: proporciona funciones específicas para trabajar en Google Colab, lo cual permite trabajar con archivos de Google Drive. Matplotlib: Biblioteca de Python, crea gráficos y visualizaciones en Python.	Simulador de brazo robótico

<p>conjunto de trazos finos en el espacio utilizando el algoritmo de Denavit-Hartenberg.</p>	<p>3. Codificación del simulador.</p>	<p>Matplotlib.pyplot: Crea gráficos estáticos y animados. mpl_toolkits.mplot3d: Sub Biblioteca de Matplotlib, crea gráficos tridimensionales. Numpy: Biblioteca de Python, usada en computación científica. Permite trabajar con matrices y operaciones numéricas en arrays multidimensionales. matplotlib.animation: Proporciona herramientas para crear animaciones en Matplotlib. Rc: configura matplotlib para mostrar animaciones en formato HTML utilizando JavaScript.</p>	
<p>Objetivo 2 Generar datasets del movimiento de trazos finos en video o coordenadas cartesianas para el entrenamiento de la cinemática inversa de un robot de 3 GDL adquiridas mediante un robot simulado.</p>	<p>1. Generar conjuntos de datos tabulares y videos e imágenes.</p>	<p>Lenguaje: Python Bibliotecas: Google.colab: proporciona funciones específicas para trabajar en Google Colab, lo cual permite trabajar con archivos de Google Drive. Pandas: herramienta que proporciona estructuras de datos y herramientas para el análisis de datos, principalmente tabulares. Numpy: Biblioteca de Python, usada en computación científica. Permite trabajar con matrices y operaciones numéricas en arrays multidimensionales. sklearn.model_selection.train_test_split: es una función de scikit-learn, con ella se divide un conjunto de datos para entrenamiento y prueba. Utilizada en el aprendizaje supervisado. Sklearn.pipeline.Pipeline: es una clase de scikit-learn, con ella se encadena múltiples pasos de procesamiento de dato y modelado en un único objeto. Sklearn.preprocessing.StandardScaler: es una clase de scikit-learn, que permite estandarizar características eliminando la media y escalando a la varianza unitaria, cuando el algoritmo funcione mejor con características en la misma escala. Sklearn.neuronal_network.MLPRegressor: es una clase de scikit-learn, que implementa una red neuronal artificial para la regresión. Con ella se predicen valores numéricos a partir de un conjunto de características de entrada. Sklearn.svm.SVR: es una clase de scikit.learn, implementa una máquina de vectores de soporte (SVM) para la regresión. Este método predice</p>	<p>Conjunto de datos tabulares Link: https://drive.google.com/drive/folders/1Xf_o_c9Qz8qP0teMaNcjLNw0RoMr3W5oR?usp=sharing Conjunto de videos (frames) Link: https://drive.google.com/drive/folders/127RmsFDj4tBSd4pZerZf9P596nZPTnkg?usp=sharing</p>

		<p>valores numéricos mediante características de entrada, construye un hiperplano en un espacio de muchas dimensiones para separar esas características en grupos.</p> <p>Sklearn.tree.DecisionTreeRegressor: es una clase de scikit-learn, implementa un árbol de decisiones para la regresión. Con ella se predice valores numéricos a partir de características de entrada, se dividen el conjunto repetidamente en subconjuntos basados en las características.</p> <p>Sklearn.ensemble.RandomForestRegressor: es una clase de scikit-learn, implementa un conjunto de árboles de decisión aleatorios para la regresión. Con ella se predicen valores numéricos a partir de características de entrada mediante la combinación de múltiples árboles de decisión para mejorar la precisión y la generalización del modelo.</p> <p>Sklearn.compose.TransformedTargetRegressor: es una clase de scikit-learn, con ella se aplica una transformación del objetivo de la regresión antes de ajustar un modelo, y de ser necesario aplica la transformación en las predicciones. Genera una interpretación más sencilla de las predicciones.</p> <p>Joblib: biblioteca de Python, que ayuda a guardar y cargar objetos, como modelos de aprendizaje automático.</p> <p>Keras.metrics.binary_crossentropy: es una función de pérdida, se usa en problemas de clasificación binaria en redes neuronales. Con ella se ajustan los parámetros del modelo y minimizar la pérdida.</p>	
<p>Objetivo 3 Comparar el compromiso entre el error y el tiempo de inferencia de un conjunto de algoritmos de Machine Learning recomendados por expertos para la selección de las mejores configuraciones en los</p>	<ol style="list-style-type: none"> 1. Experimentación con conjuntos de datos. 2. Analizar los conjuntos con modelos de Machine Learning, divididos en conjuntos de entrenamiento y validación. 	<p>Google.colab: proporciona funciones específicas para trabajar en Google Colab, lo cual permite trabajar con archivos de Google Drive.</p> <p>Cv2: una biblioteca de OpenCV en Python, se usa para el procesamiento de imágenes y visión por computadora.</p> <p>Os: biblioteca estándar de Python, proporciona funciones para interactuar con el Sistema Operativo, se usa para trabajar con archivos y directorios, para así manipularlos y así obtener información del entorno de ejecución del programa.</p> <p>Numpy: Biblioteca de Python, usada en computación científica. Permite trabajar con matrices y operaciones numéricas en arrays multidimensionales.</p>	<p>Cuadros comparativos del rendimiento de los algoritmos. Puede observarlo en la Tabla 7.</p>

<p>conjuntos de datos construidos</p>	<p>3. Entrenar los modelos, seleccionando el mejor resultado y aplicar función de pérdida para ambos conjuntos de datos.</p>	<p>Matplotlib.pyplot: biblioteca de Python, crea gráficos estáticos y animados. Keras: biblioteca de Python, se usa para entrenar redes neuronales de manera eficiente y sencilla. Proporciona un interfaz de alto nivel para definir modelos de redes neuronales, agregar capas, compilar modelos con funciones de pérdida y optimizadores, entrenando modelos con datos de entrada. Keras.layers: un módulo de Keras que proporciona una variedad de capas neuronales predefinidas, las cuales se agregan a modelo de redes neuronales, con el fin de construir diferentes tipos de arquitecturas de redes neuronales. Keras.metrics.R2Score: es un métrica de evaluación de rendimiento de un modelo de regresión. Se utiliza para evaluar que tan bien se ajustan las predicciones del modelo a los datos reales. Un valor de R cuadrado (R^2) más cercano a 1 indica un mejor ajuste del modelo.</p>	
---------------------------------------	--	---	--

Fuente. Investigación Propia.

4.2. Plan de acción detallado

En general el plan de acción esta detallado por el siguiente diagrama:

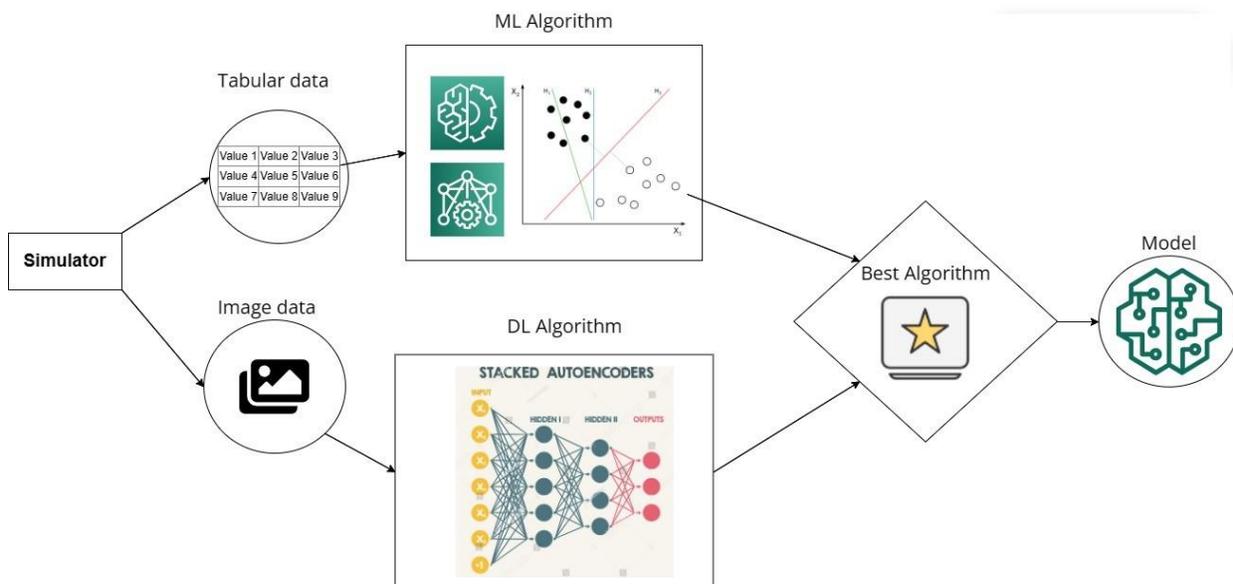


Figura 7. Diagrama de la investigación.

Fuente. Investigación Propia, Simulación brazo robótico. Diagrama de la investigación.

Para el desarrollo de los objetivos, en primera instancia se desarrolló un simulador de un brazo robótico, con el cual se produjo dos conjuntos de datos, en segunda instancia se dividieron en datos tabulares y videos e imágenes (frames) del movimiento del brazo robótico; por último, para los conjuntos de datos tabulares se usaron algoritmos de Machine Learning y para los conjuntos de video e imagen (frames) se aplicaron algoritmos de Deep Learning (Redes Neuronales Convolucionales).

A continuación, se detalla las actividades mencionadas anteriormente:

4.2.1. Objetivo 1

Para la construcción del simulador se usó la metodología en cascada, la cual es una de las metodologías más prácticas a la hora de codificar un script.

Ingeniería de requisitos

Requisitos funcionales:

1. Crear una letra:

Requisito 1.1: El sistema debe permitir al usuario definir la letra creada.

Requisito 1.2: La posición inicial de la letra y su tamaño y dirección deben ser configurables en un espacio tridimensional.

Requisito 1.3: El sistema debe poder representar letras suaves sin recortes en la animación.

2. Cinemática inversa:

Requisito 2.1: El sistema debe calcular con precisión las posiciones angulares de las articulaciones del robot para alcanzar la posición y dirección deseada en el espacio tridimensional.

Requisito 2.2: Según el autor, se deben tener en cuenta las restricciones de movimiento y las restricciones físicas al realizar cálculos de cinemática inversa.

Requisito 2.3: La interfaz debe tener la configuración de conexión correcta para ingresar las coordenadas objetivo del robot y obtener la letra.

3. Animación:

Condición 3.1: El sistema debe animar suavemente el movimiento del robot desde la posición actual hasta la posición objetivo para formar una estrella.

Condición 3.2: El usuario debe ser capaz de controlar la velocidad de la animación según sus preferencias.

Condición 3.3: Se deben proporcionar imágenes claras durante la animación para mostrar el progreso y el estado actual del sistema.

Requisitos No Funcionales

1. Rendimiento:

Condición 4.1: Para garantizar una experiencia de usuario fluida y visible, la animación debe mantener una velocidad de fotogramas adecuada.

Condición 4.2: Rendimiento inverso Los cálculos cinemáticos deben realizarse de manera eficiente para minimizar el tiempo de procesamiento y evitar retrasos notables.

2. Usabilidad:

Condición 5.1: La interfaz de usuario debe ser intuitiva y fácil de usar, con controles claros para iniciar, detener y ajustar la animación.

Condición 5.2: La documentación debe ser suficiente para que el usuario comprenda y utilice las funciones del sistema.

3. Compatibilidad:

Requisito 6.1: El sistema debe admitir versiones específicas de bibliotecas y herramientas utilizadas en la aplicación, como Matplotlib y NumPy.

Requisito 6.2: Debería poder ejecutar el sistema en entornos Python estándar sin dependencias adicionales significativas

4. Pensamiento:

Condición 7.1: El sistema debe manejar correctamente situaciones inesperadas, como desviaciones en las coordenadas de entrada o problemas de convergencia en cálculos de cinemática inversa.

Condición 7.2: Debe implementar mecanismos que proporcionen mensajes de error claros y útiles para problemas con el sistema de arranque.



Figura 8. Diagrama de Flujo Simulación Brazo robótico.

Fuente. Investigación Propia, Simulación brazo robótico. Diagrama de flujo.

Codificación

1. Importación de biblioteca

El código comienza importando las bibliotecas necesarias, como "matplotlib", "numpy", "matplotlib.animation" y "mpl_toolkits.mplot3d", que se utilizan para renderizar y mostrar animaciones gráficas en 3D.

2. Definición de funciones

Se definen varias funciones que son necesarias para que el código funcione. Estas funciones incluyen "Tz" y "Tx" para calcular transformaciones homogéneas para movimientos de traslación de los ejes z y x, "getLetter" para colocar una letra y "createLetter" para obtener coordenadas de puntos. que forman una estrella en la animación y "Cinemática inversa" calcula ángulos compuestos del robot en cinemática inversa.

3. Creación de la letra

La función "createLetter" está definida para crear las coordenadas 3D de la letra dada. La función toma como entrada una matriz que representa una estrella y devuelve un conjunto de puntos tridimensionales que forman una estrella.

4. Cinemática inversa

La función de cinemática inversa está configurada para calcular las posiciones angulares de las articulaciones del robot en cinemática inversa. La función toma como entrada las coordenadas espaciales tridimensionales del objeto y las longitudes de los eslabones del robot y devuelve las posiciones angulares correspondientes.

5. Animación

La función de actualización está configurada para actualizar la posición del robot en cada cuadro de la animación. Dentro de esta función, se calculan las posiciones angulares del robot en cinemática inversa y la animación se actualiza en consecuencia.

6. Guardar animación

Finalmente, la animación generada se guarda en un archivo de video usando la función "Guardar". Esta característica permite al usuario guardar la animación en un formato específico como MP4.

El script se elaboró en la herramienta virtual “google colab”, la cual fue un gran apoyo a la hora de presentar el simulador. se anexa el link directo al proyecto “Robot simulador”

<https://colab.research.google.com/drive/11OXJPCqqatziNs7vJqweLU9cQr4TDEdW?usp=sharing>

Pruebas

Para las pruebas del código lo que se realizó primero fue la introducción de las diferentes letras las cuales son 26, todas son ingresadas con la función “getLeter” como se ve en la figura 8 de la Simulación del Brazo Robótico (SBR)

Pseudocódigo: Ingresar coordenadas de las letras.

```
FUNCION getLeter(leter)
  SI leter == 'A' ENTONCES
    RETORNAR matriz_A
  SI leter == 'B' ENTONCES
    RETORNAR matriz_B
  SI leter == 'C' ENTONCES
    RETORNAR matriz_C
  ...
  SI leter == 'Z' ENTONCES
    RETORNAR matriz_Z
FIN FUNCION

MATRIZ matriz_A = [
  [4.0, 2.0, 0.1, 1.0, 3.0],
  [0.1, 4.0, 0.1, 2.0, 2.0],
  [0.0, 0.0, 0.0, 0.0, 0.0],
  [1.0, 1.0, 1.0, 1.0, 1.0]
]
MATRIZ matriz_B = [
  [0.1, 0.1, 3.0, 3.0, 0.1, 2.0, 2.0, 0.1],
  [5.0, 0.1, 0.0, 2.5, 2.5, 2.5, 5.0, 5.0],
  [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
  [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0]
]
...
MATRIZ matriz_Z = [
  [5.0, 0.0, 5.0, 0.0],
  [0.0, 0.0, 5.0, 5.0],
  [0.0, 0.0, 0.0, 0.0],
  [1.0, 1.0, 1.0, 1.0]
]

LEER leter
MATRIZ resultado = getLeter(leter)
IMPRIMIR resultado
```

El usuario selecciona la letra que quiere simular:

```
[36] leter=getLeter('F')
```

Figura 9. Selección de letra.

Fuente. Investigación Propia, Simulación brazo robótico. Selección de letra.

El resultado es un video en formato mp4 el cual se descargado posteriormente:

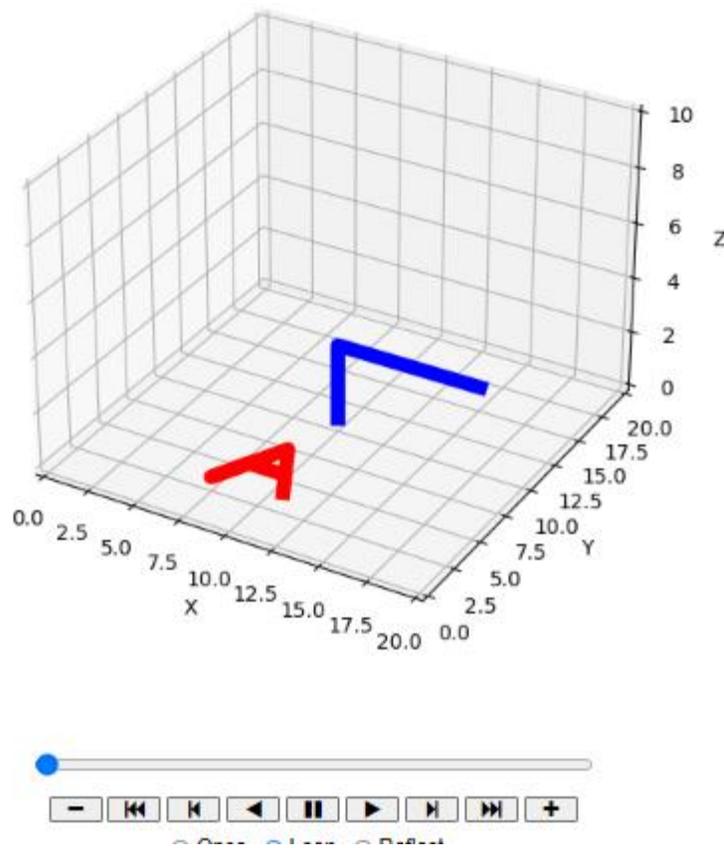


Figura 10. Video letra "A".

Fuente. Investigación Propia, Simulación brazo robótico. Video letra A.

Se anexa el link directo a una carpeta en drive donde se encuentran todas las simulaciones de las letras

<https://drive.google.com/drive/folders/1my6ESzHfGOgaH6zhQLl0fzLSbisyzShn?usp=sharing>

4.2.2. Objetivo 2

Para identificar los conjuntos de datos se aplicaron diferentes procesos, los cuales resultaron en la creación del dataset y la aplicación de los modelos para procesar los datos.

Conjunto de datos tabulares:

Para sacar las coordenadas se estructura una matriz que combina todos los archivos con las coordenadas de cada letra, siendo representadas por $x_{i1}, x_{i2}, x_{i3}, y_{i1}, y_{i2}, y_{i3}, z_{i1}, z_{i2}, z_{i3}$ Coordenada específica del eslabón del robot y la $y = [\theta_0, \theta_1, \theta_2]$ que representa los ángulos de las articulaciones, y en la tabla se muestra como q .

Tabla 3. Estructura Dataset tabulares.

ESTRUCTURA DEL DATASET																
0	q0	q1	q2	x0	x1	x2	x3	y0	y1	y2	y3	z0	z1	z2	z3	leter

Fuente. Investigación Propia.

Con base a lo anterior se realiza un modelo que analizara el movimiento de las coordenadas de cada ángulo.

Modelos

$$\hat{q}_2 = f(x, y, z, leter)$$

$$\hat{q}_1 = f(\hat{q}_2, x, y, z, leter)$$

$$\hat{q}_0 = f(\hat{q}_1, \hat{q}_2, x, y, z, leter)$$

Los anteriores modelos permiten establecer relaciones matemáticas en las variables de entrada y los valores objetivo (q_0, q_1, q_2), utilizando conocimiento previo para realizar las predicciones

sobre estos valores en un nuevo conjunto de datos. Estos modelos fueron utilizados para determinar la capacidad para predecir y estimar valores desconocidos.

A ese nuevo conjunto de datos se realiza un análisis comparativo de varios modelos de aprendizaje utilizando diferentes tamaños de ventanas.

El algoritmo de ventanas permite adaptar los modelos a diferentes partes de una serie temporal. Consiste en dividir una serie temporal en varias secciones o “Ventanas” de tamaño fijo y se aplicó un modelo de Machine Learning a cada ventana, en este caso, se aplicaron ventanas de [1,2,4,8] a un conjunto de datos dividido en datos de entrenamiento 80% y prueba 20%, donde 1063 muestras fueron de entrenamiento y 263 muestras fueron de prueba.

A partir de lo anterior con ayuda de los modelos de aprendizaje automático Maquina como lo son Soporte de Vectores (SVM), Redes Neuronales Artificiales (ANN), Arboles de Decision (DT), y Bosques Aleatorios (RF), utilizando los datos de entrenamiento de escalados. Luego, Se realizaron predicciones en los datos de prueba y se evalúa el rendimiento de cada modelo utilizando métricas como el coeficiente de determinación (R^2) y la función de pérdida binaria cruzada (Binary Crossentropy) como se puede observar en la siguiente tabla:

Tabla 4. Dataset Pruebas para series temporales Multivariantes.

Modelo	Ventana	Coeficiente determinación (R^2)		Perdida Binaria Cruzada (BCE)	
		Entrenamiento (train)	Prueba (test)	Entrenamiento (train)	Prueba (test)
Svm	1	0.9861	0.9677	0.6158	0.6044
Ann	1	0.9971	0.9970	0.6147	0.6032
Dt	1	1.0	0.9964	0.6143	0.6031
Rf	1	0.9988	0.9984	0.6145	0.6030
Svm	2	0.9865	0.9690	0.6157	0.6043
Ann	2	0.9980	0.9973	0.6145	0.6031
Dt	2	1.0	0.9970	0.6142	0.6031
Rf	2	0.9989	0.9987	0.6144	0.6030
Svm	4	0.9889	0.9737	0.6153	0.6043
Ann	4	0.9965	0.9970	0.614	0.6032
Dt	4	1.0	0.9984	0.6141	0.6030
Rf	4	0.9988	0.9989	0.6143	0.6030
Svm	8	0.9830	0.9563	0.6153	0.6058
Ann	8	0.9969	0.9840	0.6140	0.6042
Dt	8	1.0	0.9966	0.6137	0.6036
Rf	8	0.9987	0.9986	0.6139	0.6036

Fuente. Investigación Propia.

4.2.3. Objetivo 3

Entendimiento de los datos

Lo primero que se hizo fue cargar una serie de archivos CSV de los datos tabulares de cada letra desde una carpeta en drive. Se combinan los datos de estos archivos en un solo DataFrame. Después define un mapeo de letras a números y una función para aplicar este mapeo a una columna de letras en el DataFrame. Reemplazando las letras por sus equivalentes numéricos en una columna específica del DataFrame. Finalmente, guarda el DataFrame modificado en un nuevo archivo CSV. Como se puede apreciar en el siguiente diagrama.

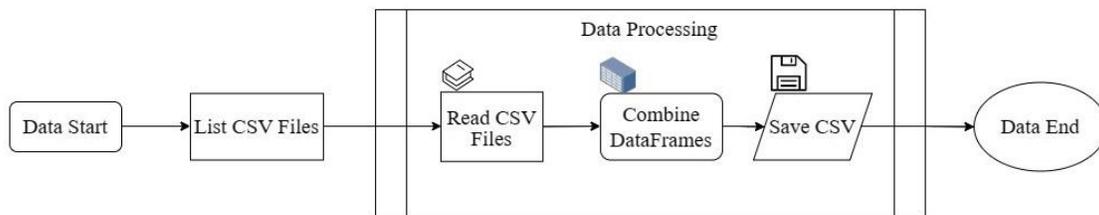


Figura 11. Carga de DATAFRAME.

Fuente. Investigación Propia, Pruebas Dataset Tabulares. Carga DATEFRAME

Modelamiento

Para este punto ya se tiene presente los diferentes datos tanto tabulares como de video, para el uso de los modelos se hizo una partición de los datos, como se presenta en la siguiente figura:

```
Xtrain,Xtest,ytrain,ytest=train_test_split(X,y,test_size=0.2,random_state=2023)
```

Figura 12. Partición Dataset.

Fuente. Investigación Propia, pruebas Dataset Tabulares. Partición Dataset.

Esta línea de código divide el conjunto de datos en conjuntos de entrenamiento y prueba, tanto para las características como para las etiquetas, con el 80% de los datos para entrenamiento y el 20% para prueba, asegurando que esta división sea reproducible con una semilla específica.

Uso de las ventanas

Este código ejecuta un bucle que itera sobre una lista de valores de rango(ventanas). En cada iteración, se genera un conjunto de datos de entrenamiento y prueba a partir de un conjunto de datos base `X, y` utilizando la función `getDataset`. Luego, normaliza las etiquetas y entrena varios modelos de regresión, incluyendo SVM, redes neuronales artificiales, árbol de decisión y bosque aleatorio, utilizando los datos de entrenamiento. Los modelos se evalúan en los conjuntos de entrenamiento y prueba, y se registran los resultados. Finalmente, se almacenan todos los resultados en una lista llamada `results` para su posterior análisis. el código puede ser visualizado en la siguiente figura:

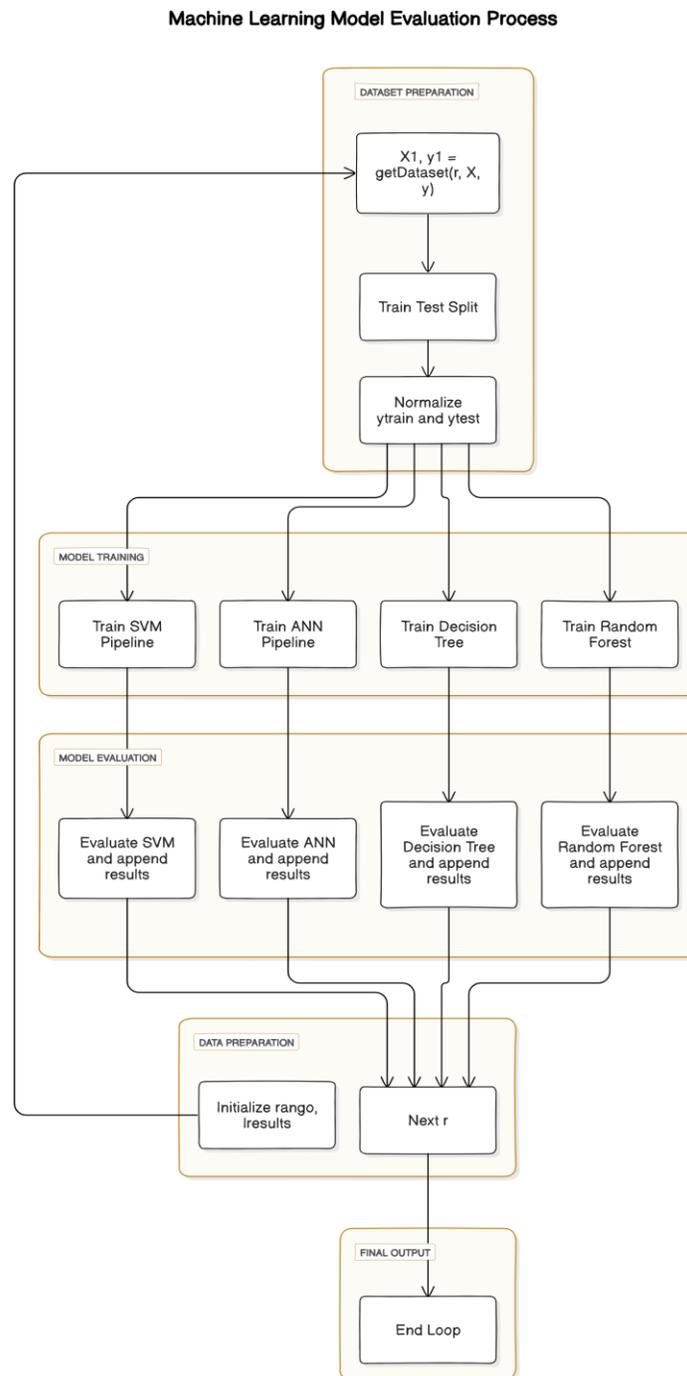


Figura 13. Patrones series temporales multivariantes.

Fuente. Investigación Propia. Pruebas Dataset Tabulares. Patrones series temporales multivariantes.

En cada iteración del bucle, se genera un conjunto de datos de entrenamiento y prueba a partir de un conjunto de datos base, luego se normalizan las etiquetas y se entrenan varios modelos de regresión, incluyendo Support Vector Machine (SVM), Redes Neuronales Artificiales (ANN), Árbol de Decisión (DT) y Bosque Aleatorio (RF). Después de entrenar cada modelo, se evalúan sus rendimientos en los conjuntos de entrenamiento y prueba, y se registran los resultados. Esto permite comparar el desempeño de los diferentes modelos para diferentes rangos de valores.

Se usa este rango de ventanas ya que si bien usar más dará un resultado más exacto se correría el riesgo de que los datos sufran un “overfitting” o sobreajuste, un fenómeno en el aprendizaje automático donde un modelo se adapta demasiado bien a los datos de entrenamiento, lo que resulta en un rendimiento deficiente en datos nuevos o no vistos.

Resultados de los datos tabulares

Los resultados proporcionados son las métricas de rendimiento de varios modelos de regresión entrenados para diferentes valores de un parámetro (representado por 'r') en el rango de 1, 2, 4 y 8. Ahora, se observan los resultados para cada modelo y cada valor de 'r':

1. Para el modelo SVM (Support Vector Machine)

Se observa un alto R cuadrado tanto en los conjuntos de entrenamiento como en los de prueba, lo que indica un buen ajuste del modelo a los datos.

Los valores de binary crossentropy son bajos, lo que sugiere un buen rendimiento en la tarea de clasificación binaria.

2. Para el modelo ANN (Redes Neuronales Artificiales)

Similar al modelo SVM, se obtienen altos valores de R cuadrado tanto en entrenamiento como en prueba.

Los valores de binary crossentropy también son bajos, indicando un buen rendimiento en la tarea de clasificación binaria.

3. Para los modelos DT (Árbol de Decisión) y RF (Bosque Aleatorio)

Se obtienen valores de R cuadrado cercanos a 1 en ambos conjuntos de datos para cada valor de 'r', lo que indica un ajuste casi perfecto.

Los valores de binary crossentropy son bajos, lo que sugiere un buen rendimiento en la tarea de clasificación binaria.

En general, los modelos parecen tener un buen rendimiento en la tarea de predicción y clasificación binaria para los diferentes valores de 'r'. Sin embargo, es importante tener en cuenta que el mejor modelo puede depender del contexto específico del problema y de otras consideraciones como el costo computacional y la interpretabilidad del modelo.

A continuación, se adjunta el link del script utilizado para obtener patrones con series temporales multivariantes:

<https://drive.google.com/file/d/1bgHWnjN0NtpgybmtevMpjWfzksfqDrdG/view?usp=sharing>

Resultados Dataset imágenes

El código comienza leyendo videos de un directorio especificado, redimensionando los frames de los videos y alimentándose en una estructura de datos. Luego, aplica una técnica de aumento de la resolución temporal a los videos para mejorar la calidad de los datos. Posteriormente, divide el conjunto de datos en entrenamiento y validación para su uso en el proceso de entrenamiento del modelo.

Después, prepara los datos para el modelo creando pares de frames desplazados, que servirán como entrada y salida para el modelo. Finalmente, define y entrena un modelo de red neuronal ConvLSTM utilizando los datos preparados, con el objetivo de predecir frames futuros en secuencias de video. El modelo se compila con una función de pérdida y un optimizador específicos, y se entrena utilizando callbacks para mejorar el proceso de entrenamiento. Para el entrenamiento se usó 5 épocas, como se observa en la siguiente figura:

```
Epoch 1/5
4/4 [=====] - 262s 65s/step - loss: 0.6342 - val_loss: 0.6665 - lr: 0.0010
Epoch 2/5
4/4 [=====] - 254s 65s/step - loss: 0.4268 - val_loss: 0.5662 - lr: 0.0010
Epoch 3/5
4/4 [=====] - 253s 65s/step - loss: 0.2375 - val_loss: 0.5163 - lr: 0.0010
Epoch 4/5
4/4 [=====] - 250s 64s/step - loss: 0.2068 - val_loss: 0.4565 - lr: 0.0010
Epoch 5/5
4/4 [=====] - 259s 66s/step - loss: 0.2088 - val_loss: 0.3685 - lr: 0.0010
```

Figura 14. Pruebas Dataset imágenes con épocas.

Fuente. Investigación Propia. Pruebas Dataset imágenes. Entrenamiento en épocas.

Los resultados proporcionan una visión detallada del progreso del entrenamiento del modelo a lo largo de múltiples épocas. Cada época representa una iteración completa a través de todo el conjunto de datos de entrenamiento. La métrica de pérdida (`loss`) muestra cuánto se desvían las predicciones del modelo de los valores reales durante el entrenamiento, mientras que la pérdida de validación (`val_loss`) proporciona una medida similar en el conjunto de datos de validación, que no se utiliza para entrenar el modelo.

Se puede observar cómo ambas métricas disminuyen gradualmente a lo largo de las épocas. Este descenso indica que el modelo está aprendiendo y mejorando su capacidad para hacer predicciones precisas a medida que se expone a más datos y ciclos de entrenamiento.

Se anexa el script donde se hicieron las pruebas:

https://colab.research.google.com/drive/1OHFXAoAxM5WXTnV9yWZvU_eMLSwXurQZ?usp=sharing

5. ANÁLISIS Y DISCUSIÓN DE RESULTADOS

5.1. *Análisis*

Durante el desarrollo del proyecto se generaron 2 dataset, conformados por datos Tabulares, y no tabulares(videos), los primeros son conformados por matrices numéricas, que contienen las coordenadas cartesianas del movimiento del brazo robótico y los cálculos de estos tras la

realización de las transformaciones y los segundos son frames o datos no tabulares, conformados por videos de la simulación del brazo robótico.

El simulador ofrece la capacidad de recopilar trazos de cualquier tipo siempre y cuando se proporcionen las coordenadas de las letras. Este enfoque brinda una flexibilidad considerable al permitir la recopilación de trazos que se pueden obtener mediante los datos tabulares y de vídeo.

Para la construcción del dataset, se llevó a cabo una evaluación exhaustiva de los algoritmos utilizados, los cuales demostraron un rendimiento sobresaliente con puntuaciones de precisión por encima de 0,9; esta alta precisión proporciona una sólida base para la construcción del dataset, garantizando que los datos recopilados sean de alta calidad y adecuados para su uso en aplicaciones de aprendizaje automático y análisis de datos.

Los resultados muestran el rendimiento de diferentes modelos de regresión después de ser transformados utilizando el mismo método de escala (StandardScaler) y luego aplicar TransformedTargetRegressor para manejar la variable objetivo transformada. Aquí está un análisis de estos resultados:

1. Redes Neuronales Artificiales (ANN)

El modelo ANN muestra un alto coeficiente de determinación (R cuadrado) en el conjunto de prueba, con un valor de aproximadamente 0.998. Esto indica que el modelo ANN explica bien la variabilidad de los datos de prueba.

Utilizar la transformación del objetivo con TransformedTargetRegressor parece haber mejorado el rendimiento del modelo, ya que el valor del R cuadrado es más alto que el valor típico obtenido sin esta transformación.

```
annp=Pipeline([('scaler',StandardScaler()),('ANN',MLPRegressor())])
annpt=TransformedTargetRegressor(transformer=StandardScaler(),regressor=annp)
annpt.fit(xtrain,ytrain)
annpt.score(xtest,ytest)

0.997716148608841
```

Figura 15. Resultados con modelo ANN.

Fuente. Investigación Propia. Pruebas Dataset tabulares. Entrenamiento Dataset tabular con ANN.

2. Máquinas de Vectores de Soporte (SVM)

El modelo SVM tiene un coeficiente de determinación (R cuadrado) ligeramente menor en el conjunto de prueba, con un valor de aproximadamente 0.995.

La transformación del objetivo no parece haber tenido un impacto significativo en el rendimiento del modelo SVM.

```
svm=Pipeline([('Scaler',StandardScaler()),('SVM',SVR())])
svm=TransformedTargetRegressor(transformer=StandardScaler(),regressor=svm)
svm.fit(Xtrain,ytrain)
svm.score(Xtest,ytest)

0.9951767392761016
```

Figura 16. Resultados con modelo SVM.

Fuente. Investigación Propia. Pruebas Dataset tabulares. Entrenamiento Dataset tabular con SVM.

3. Árbol de Decisión (DT)

El modelo DT muestra un coeficiente de determinación (R cuadrado) similar al del SVM, con un valor de aproximadamente 0.995 en el conjunto de prueba.

Al igual que el SVM, la transformación del objetivo no parece haber tenido un impacto significativo en el rendimiento del modelo DT.

```
dt=Pipeline([('Scaler',StandardScaler()),('DecisionTree',DecisionTreeRegressor())])
dt=TransformedTargetRegressor(transformer=StandardScaler(),regressor=dt)
dt.fit(Xtrain,ytrain)
dt.score(Xtest,ytest)

0.995185703606436
```

Figura 17. Resultados con Modelo DT.

Fuente. Investigación Propia. Pruebas Dataset tabulares. Entrenamiento Dataset tabular con SVM.

4. Bosque Aleatorio (RF)

El modelo RF muestra el mejor rendimiento de todos los modelos, con un coeficiente de determinación (R cuadrado) en el conjunto de prueba de aproximadamente 0.999.

La transformación del objetivo también parece haber mejorado el rendimiento del modelo RF.

```
rf=Pipeline([('Scaler',StandardScaler()),('RandomForest',RandomForestRegressor())])
rf=TransformedTargetRegressor(transformer=StandardScaler(),regressor=rf)
rf.fit(xtrain,ytrain)
rf.score(xtest,ytest)

0.9986029193825083
```

Figura 18. Resultados con modelo RF.

Fuente. Investigación Propia. Pruebas Dataset tabulares. Entrenamiento Datase tabular con RF.

En resumen, la transformación del objetivo utilizando TransformedTargetRegressor ha mejorado el rendimiento de algunos modelos, especialmente en el caso de las Redes Neuronales Artificiales (ANN) y el Bosque Aleatorio (RF). Sin embargo, el impacto de esta transformación puede variar según el modelo y el conjunto de datos específicos.

En cuanto a los resultados de los datos tabulares se obtuvo que los modelos de regresión entrenados muestran un rendimiento generalmente alto y robusto en la tarea de predicción, tanto en los conjuntos de entrenamiento como en los de prueba. Esto sugiere que los modelos son capaces de capturar efectivamente la relación entre las características y la variable objetivo en los datos de entrenamiento, y también generalizan bien a datos nuevos o no vistos. Este alto rendimiento es evidenciado por los altos valores de coeficiente de determinación (R cuadrado) y los bajos valores de binary crossentropy (bce) en ambos conjuntos de datos. Como se observa en la *“Tabla 4 Dataset Pruebas para series temporales Multivariantes.”*, adjuntada a continuación:

Modelo	Ventana	Coeficiente determinación (R^2)		Perdida Binaria Cruzada (BCE)	
		Entrenamiento (train)	Prueba (test)	Entrenamiento (train)	Prueba (test)
Svm	1	0.9861	0.9677	0.6158	0.6044
Ann	1	0.9971	0.9970	0.6147	0.6032
Dt	1	1.0	0.9964	0.6143	0.6031

Rf	1	0.9988	0.9984	0.6145	0.6030
Svm	2	0.9865	0.9690	0.6157	0.6043
Ann	2	0.9980	0.9973	0.6145	0.6031
Dt	2	1.0	0.9970	0.6142	0.6031
Rf	2	0.9989	0.9987	0.6144	0.6030
Svm	4	0.9889	0.9737	0.6153	0.6043
Ann	4	0.9965	0.9970	0.614	0.6032
Dt	4	1.0	0.9984	0.6141	0.6030
Rf	4	0.9988	0.9989	0.6143	0.6030
Svm	8	0.9830	0.9563	0.6153	0.6058
Ann	8	0.9969	0.9840	0.6140	0.6042
Dt	8	1.0	0.9966	0.6137	0.6036
Rf	8	0.9987	0.9986	0.6139	0.6036

Fuente. Investigación Propia.

Mientras que los resultados del entrenamiento con video muestran una mejora gradual en la capacidad predictiva del modelo a medida que avanzan las épocas. Como se observa en la siguiente tabla:

Tabla 5. Entrenamiento Red Neuronal con Videos.

Época (Epoch)	Tiempo/Paso	Train Loss	Validation Loss (Val_loss)	Leraning Rate (lr)
1	262s 65s/step	0.6342	0.6665	0.0010
2	254s 65s/step	0.4268	0.5662	0.0010
3	253s 65s/step	0.2375	0.5163	0.0010
4	250s 64s/step	0.2068	0.4565	0.0010
5	259s 66s/step	0.2088	0.3685	0.0010

Fuente. Investigación Propia

La disminución de la pérdida tanto en el conjunto de entrenamiento como en el de validación indica que el modelo está aprendiendo de manera efectiva y generalizando bien a datos no vistos durante el entrenamiento. Además, la evaluación del modelo en el conjunto de prueba arroja una pérdida de aproximadamente 0.3685. Esta pérdida representa la discrepancia promedio entre las predicciones del modelo y las etiquetas reales en el conjunto de prueba.

En este caso, una pérdida de 0.3685 sugiere que el modelo tiene un buen rendimiento en el conjunto de prueba y es efectivo para la tarea para la que fue entrenado.

Tabla 6. Resultados del conjunto de validación.

Paso evaluación	Tiempo/Paso	Perdida en el conjunto de validación (Loss)
1	20s	0.3685

Fuente. Extraído de Investigación propia. Resultados Conjunto de validación.

Se adjunta imagen del resulta en el script:

```
# Evaluar el modelo en el conjunto de prueba
evaluation_results = model.evaluate(x_val, y_val)

# Imprimir los resultados de la evaluación (pérdida y métricas)
print("Loss:", evaluation_results)

1/1 [=====] - 20s 20s/step - loss: 0.3685
Loss: 0.3684936463832855
```

Figura 19. Función de perdida Dataset de imágenes.

Fuente. Investigación Propia. Pruebas Dataset imágenes. Función de Perdida.

En la siguiente tabla se puede observar una clara diferencia arrojada para los conjuntos de datos tabulares y de video e imagen, esto nos permite concluir que se cumplió lo propuesto en la hipótesis investigativa, la función de perdida para ambos tipos de datos nos arroja un rango mayo para Datos Tabulares y un rango mejor para Datos Videos e imágenes. Como se observa en la siguiente tabla:

Tabla 7. Comparación función de Perdida para Datasets.

Valor en pérdida Dataset Tabulares (BCE)	Valor pérdida Dataset Frames (R^2)
0.6158007791739671	0.3684936463832855

Fuente. Extraído de Investigación propia. Comparación de valores de función de Perdida para Datasets.

5.2. Discusión

Con base a los antecedentes Internacionales, podemos identificar que se generó un Conjunto de datos tabulares y de video e imágenes, con el fin de proporcionar para futuras investigaciones un

grupo de datos que puedan ser usados para experimentos de aprendizaje automático (ML) y aprendizaje profundo (DL), que estuviera formado por datos sencillos de interpretar y que permitan un procesamiento óptimo.

También es importante destacar, el desarrollo del simulador del brazo robótico de 3 grados de libertad, lo cual proporciona una vista dinámica de la cinemática del mismo, los resultados obtenidos con la aplicación de modelos de aprendizaje automático (ML) y de aprendizaje profundo (DL) pueden servir de apoyo a la asacases de desarrollo local de este tipo de datos, puesto que, en la investigación, los antecedentes regionales no abundan proyectos similares.

6. CONCLUSIONES

Al realizar un análisis de la información obtenida en esta investigación, es posible observar que se llevó a cabo la realización del simulador del movimiento de un brazo robótico de 3 grados de libertad, al igual que el dataset de los trazos finos en el espacio, que para este proyecto fue de las letras del abecedario en mayúsculas, con el fin de agilizar su realización.

De esta investigación se desarrollaron las siguientes conclusiones:

1. El desarrollo del simulador de un brazo robótico de 3 GDL que resuelva un conjunto de trazos finos utilizando el algoritmo de Denavit – Hartenberg fue exitosa, esto proporciona un modelo disponible para investigaciones futuras y una herramienta para el desarrollo de las mismas, además, se generaron los dos conjuntos de datos que permitieron el entrenamiento de con los modelos de aprendizaje automático.
2. En vista de que la generación de datos es crucial para garantizar la calidad y fiabilidad de los mismos, los conjuntos de datos utilizados permitieron recopilar información relevante en esta investigación, con ellos se establecieron la base necesaria para entrenar y evaluar los modelos de aprendizaje automático, lo que proveyó resultados significativos en esta investigación.
3. Con base a los resultados obtenidos mediante el uso de un brazo robótico simulado de 3 grados de libertad, se puede concluir que para los movimientos ejecutados utilizando técnicas de Machine Learning con conjuntos de datos como videos e imágenes exhiben un margen de error igual o menor al de un robot entrenado con datos tabulares. Este hallazgo

respalda la hipótesis investigativa, demostrado que al utilizar un dataset de frames se mejora la precisión y la eficacia de los modelos de aprendizaje automático en la cinemática de un brazo robótico.

Bibliografía

- [1] K. Grochow, S. L. Martin, A. Hertzmann, y Z. Popović, “Style-based inverse kinematics”, *ACM Trans Graph*, vol. 23, núm. 3, pp. 522–531, ago. 2004, doi: 10.1145/1015706.1015755.
- [2] H. Ren y P. Ben-Tzvi, “Learning inverse kinematics and dynamics of a robotic manipulator using generative adversarial networks”, *Rob Auton Syst*, vol. 124, p. 103386, feb. 2020, doi: 10.1016/j.robot.2019.103386.
- [3] J. S. Toquica, P. S. Oliveira, W. S. R. Souza, J. M. S. T. Motta, y D. L. Borges, “An analytical and a Deep Learning model for solving the inverse kinematic problem of an industrial parallel robot”, *Comput Ind Eng*, vol. 151, p. 106682, ene. 2021, doi: 10.1016/j.cie.2020.106682.
- [4] A. Kaplan y M. Haenlein, “Siri, Siri, in my hand: Who’s the fairest in the land? On the interpretations, illustrations, and implications of artificial intelligence”, *Bus Horiz*, vol. 62, núm. 1, pp. 15–25, ene. 2019, doi: 10.1016/j.bushor.2018.08.004.
- [5] M. Pozo Benitez, “AWS: El proveedor de Cloud para Startups”, *Universidad de Sevilla. Grado en Finanzas y Contabilidad*, pp. 21–25, 2021, Consultado: el 7 de abril de 2023. [En línea]. Disponible en: <https://idus.us.es/handle/11441/127987>
- [6] S. Vakilinia, M. M. Ali, y D. Qiu, “Modeling of the resource allocation in cloud computing centers”, *Computer Networks*, vol. 91, pp. 453–470, nov. 2015, doi: 10.1016/j.comnet.2015.08.030.
- [7] A. Sabharwal y B. Selman, “Book review”, *Artif Intell*, vol. 175, núm. 5–6, pp. 935–937, abr. 2011, doi: 10.1016/j.artint.2011.01.005.
- [8] Y. Bengio, A. Courville, y P. Vincent, “Representation Learning: A Review and New Perspectives”, jun. 2012, Consultado: el 7 de abril de 2023. [En línea]. Disponible en: <https://arxiv.org/pdf/1206.5538.pdf>
- [9] J. Kelleher, “Deep Learning”, en *MIT Press Essential Knowledge series*, 2019, pp. 1–4. Consultado: el 7 de abril de 2023. [En línea]. Disponible en:

- https://books.google.com.co/books?id=b06qDwAAQBAJ&dq=deep+learning&lr=&hl=es&source=gbs_navlinks_s
- [10] A. Hernández, Maria. Becerra, Brenda. Dubaln, Gloria. Mata, Emanuel. Morales, y A. Franco, “Diseño y fabricación de robot didáctico programable de 3 grados de libertad”, *Pistas Educativas*, vol. 137, pp. 296–298, nov. 2020, Consultado: el 7 de abril de 2023. [En línea]. Disponible en: <https://pistaseducativas.celaya.tecnm.mx/index.php/pistas/article/view/2288/1835>
- [11] V. Santamaria, “Diseño, montaje, implementación y simulación de un brazo robótico con 3 grados de libertad con fines educativos.”, *Universidad Politécnica de Valencia*, pp. 9–14, 2020, Consultado: el 7 de abril de 2023. [En línea]. Disponible en: <https://riunet.upv.es/bitstream/handle/10251/153275/Ferrando%20-%20Dise%c3%b1o%2c%20montaje%2c%20implementaci%c3%b3n%20y%20simulaci%c3%b3n%20de%20un%20brazo%20rob%c3%b3tico%20con%203%20grados%20de%20libe....pdf?>
- [12] J. Neyra, “Diseño e implementación de las funciones de agarre y levante en un brazo kinova usando señales EEG y Deep Learning”, *Repositorio Digital de Tesis y Trabajos de Investigación PUCP*, 2020, Consultado: el 7 de abril de 2023. [En línea]. Disponible en: <http://hdl.handle.net/20.500.12404/17104>
- [13] D. Gómez y Á. Bustos, “Desarrollo matemático y aplicación de la cinemática inversa por medio de un software de cálculo y programación, para el robot Mitsubishi rv-m1”, *Repositorio Institucional Universidad Distrital - RIUD*, 2018, Consultado: el 7 de abril de 2023. [En línea]. Disponible en: <https://repository.udistrital.edu.co/bitstream/handle/11349/13402/G%c3%b3mezCamposDar%c3%adoAndr%c3%a9s2018.pdf?sequence=1&isAllowed=y>
- [14] D. Páez Ramírez, J. P. Romero Camacho, y J. G. Guarnizo Marin, *UR3 modelo cinemático inverso*. Universidad Santo Tomás, 2021. doi: 10.15332/dt.inv.2021.02849.

- [15] L. Navarro y J. Avalos, “DidaPlat: Plataforma robótica de bajo costo con 3 GDL para fines educativos”, vol. 12, p. 11, 2019, Consultado: el 7 de abril de 2023. [En línea]. Disponible en: <https://publicaciones.uci.cu/index.php/serie/article/view/497>
- [16] H. Belalcazar, J. Botina, y A. Zambrano, “Diseño y construcción de un prototipo de robot paralelo orientado a procesos siderúrgicos de corte de lámina, a través del acondicionamiento de equipo de oxicorte, en la ciudad de san juan de pasto”, *Repositorio Institucional - Corporación Universitaria Autónoma de Nariño*, dic. 2015, Consultado: el 7 de abril de 2023. [En línea]. Disponible en: <http://repositorio.aunar.edu.co:8080/xmlui/bitstream/handle/20.500.12276/574/IE89A.pdf?sequence=5&isAllowed=y>
- [17] J. Castellanos-Ruíz, L. M. Montealegre-Mesa, B. D. Martínez-Toro, J. J. Gallo-Serna, y O. Almanza-Fuentes, “Uso de sensores inerciales en fisioterapia: Una aproximación a procesos de evaluación del movimiento humano”, *Univ Salud*, vol. 23, núm. 1, pp. 55–63, dic. 2020, doi: 10.22267/rus.212301.214.
- [18] T. Nguyen, “The Effectiveness of Online Learning: Beyond No Significant Difference and Future Horizons”, *J Online Learn Teach*, vol. 11, pp. 309–319, jun. 2015, Consultado: el 7 de abril de 2023. [En línea]. Disponible en: https://jolt.merlot.org/Vol11no2/Nguyen_0615.pdf
- [19] IBM Documentation Help, “Datos tabulares frente a datos transaccionales”, IBM Documentation Help.
- [20] M. Gómez, “Cinemática Inversa”, es.scribd.
- [21] U. von Luxburg, R. Williamson, y I. Guyon, “Clustering: Science or Art?”, *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, vol. 27, pp. 65–79, jul. 2012, Consultado: el 7 de abril de 2023. [En línea]. Disponible en: <http://proceedings.mlr.press/v27/luxburg12a/luxburg12a.pdf>
- [22] C. Conejo, “Brazo Robótico de 3 Grados de Libertad con Inteligencia Artificial”. Consultado: el 8 de abril de 2023. [En línea]. Disponible en: https://www.academia.edu/37382249/Brazo_Rob%C3%B3tico_de_3_Grados_de_Libertad_con_Inteligencia_Artificial

- [23] J. A. G., . A. F., y . J. D., “Estudio y Modelado de un Brazo Robótico Lego de tres grados de libertad”, *Lámpsakos*, pp. 5–11, 2011, [En línea]. Disponible en: <https://www.redalyc.org/articulo.oa?id=613965338002>
- [24] I. Gómez, “Introducción al aprendizaje supervisado e implementación de una red neuronal en Python”, *Repositorio Universitat Jaume I*, oct. 2018.
- [25] P. Avitia Carlos, B. Rodríguez Tapia, y N. Candolfi Arballo, “El uso de simuladores para la enseñanza en línea de electrónica básica”, en *Proceedings INNODOCT/21. International Conference on Innovation, Documentation and Education*, Valencia: Editorial Universitat Politècnica de València, oct. 2021. doi: 10.4995/INN2021.2021.13364.
- [26] Octavio, “Sistema de coordenadas Cartesianas ”, *UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO*, pp. 1–8, abr. 2011.
- [27] P. González Casimiro, “Error cuadrático medio de predicción para modelos estructurales de series temporales”, *Estadística española, ISSN 0014-1151, N° 129, 1992, págs. 117-136*, núm. 129, pp. 117–136, 1992.
- [28] E. Martínez, “Errores frecuentes en la interpretación del coeficiente de determinación lineal”, *Anuario Jurídico y Económico Escurialense*, vol. 38, pp. 315–332, 2005, Consultado: el 9 de abril de 2023. [En línea]. Disponible en: <https://docplayer.es/21127540-Errores-frecuentes-en-la-interpretacion-del-coeficiente-de-determinacion-lineal.html>
- [29] H. Li, J. Liu, Z. Yang, R. W. Liu, K. Wu, y Y. Wan, “Adaptively constrained dynamic time warping for time series classification and clustering”, *Inf Sci (N Y)*, vol. 534, pp. 97–116, sep. 2020, doi: 10.1016/j.ins.2020.04.009.
- [30] K. Ramírez, “Cinemática Directa del Robot”, [kramirez.net](http://www.kramirez.net). Consultado: el 8 de abril de 2023. [En línea]. Disponible en: <http://www.kramirez.net/wp-content/uploads/2012/04/CinematicaDirectaRobot.pdf>
- [31] B. Schölkopf y A. J. Smola, *Learning with Kernels*. The MIT Press, 2018. doi: 10.7551/mitpress/4175.001.0001.

- [32] C. J. C. Burges, “A Tutorial on Support Vector Machines for Pattern Recognition”, *Data Min Knowl Discov*, vol. 2, núm. 2, pp. 121–167, 1998, doi: 10.1023/A:1009715923555.
- [33] J. Heaton, “Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning”, *Genet Program Evolvable Mach*, vol. 19, núm. 1–2, pp. 305–307, jun. 2018, doi: 10.1007/s10710-017-9314-z.
- [34] S. L. Salzberg, “C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993”, *Mach Learn*, vol. 16, núm. 3, pp. 235–240, sep. 1994, doi: 10.1007/BF00993309.
- [35] J. R. Quinlan, “Induction of decision trees”, *Mach Learn*, vol. 1, núm. 1, pp. 81–106, mar. 1986, doi: 10.1007/BF00116251.
- [36] T. Hastie, R. Tibshirani, y J. Friedman, *The Elements of Statistical Learning*. New York, NY: Springer New York, 2009. doi: 10.1007/978-0-387-84858-7.
- [37] L. Breiman, “Random Forests. ”, *Mach Learn*, vol. 45, núm. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.
- [38] L. Cuenya y E. Ruetti, “EPISTEMOLOGICAL AND METHODOLOGICAL CONTROVERSIES BETWEEN THE QUALITATIVE AND QUANTITATIVE PARADIGM IN PSYCHOLOGY”, *Revista Colombiana de Psicología*, vol. 19, núm. 2, pp. 271–277, jul. 2010, [En línea]. Disponible en: <https://revistas.unal.edu.co/index.php/psicologia/article/view/17795>
- [39] R. R. Suarez, “El paradigma Positivista en la investigación”. el 1 de julio de 2010.
- [40] M. de la L. Casas Martínez, “INTRODUCCIÓN A LA METODOLOGÍA DE LA INVESTIGACIÓN EN BIOÉTICA: SUGERENCIAS PARA EL DESARROLLO DE UN PROTOCOLO DE INVESTIGACIÓN CUALITATIVA INTERDISCIPLINARIA”, *Acta Bioeth*, vol. 14, núm. 1, 2008, doi: 10.4067/S1726-569X2008000100013.
- [41] F. Anselmo y S. Flores, “Fundamentos Epistémicos de la Investigación Cualitativa y Cuantitativa: Consensos y Disensos Epistemic Fundamentals of Qualitative and Quantitative Research: Consensus and Dissensus Fundamentos epistémicos da

- pesquisa qualitativa e quantitativa: consensos e dissensos”, vol. 13, núm. 1, pp. 102–122, 2019, Consultado: el 19 de noviembre de 2023. [En línea]. Disponible en: <http://www.scielo.org.pe/pdf/ridu/v13n1/a08v13n1.pdf>
- [42] F. Gilbes, “El Método Científico”, *bioanatomia*, 2020, Consultado: el 9 de abril de 2023. [En línea]. Disponible en: https://bioanatomia.edu.pe/wp-content/uploads/2020/10/metodo_cientifico.pdf
- [43] M. Burge, “La ciencia. Su método y su filosofía”, dic. 2018, Consultado: el 9 de abril de 2023. [En línea]. Disponible en: https://users.dcc.uchile.cl/~cgutierrez/cursos/INV/bunge_ciencia.pdf
- [44] C. Villegas, “Investigación correlacional”, *IDCOPUB*, 2011, Consultado: el 9 de abril de 2023. [En línea]. Disponible en: <https://idoc.pub/documents/investigacion-correlacional-5143x5ow8v4j>
- [45] C. Martínez, “Investigación Descriptiva: Tipos y Características”.
- [46] R. O. Tinoco, E. B. Goldstein, y G. Coco, “A data-driven approach to develop physically sound predictors: Application to depth-averaged velocities on flows through submerged arrays of rigid cylinders”, *Water Resour Res*, vol. 51, núm. 2, pp. 1247–1263, feb. 2015, doi: 10.1002/2014WR016380.
- [47] J. Villagrán, “Técnicas de recopilación de datos que sí funcionan”, Data Scope. Consultado: el 19 de noviembre de 2023. [En línea]. Disponible en: <https://datascope.io/es/blog/todo-sobre-las-tecnicas-de-recopilacion-de-datos/#:~:text=Todo%20sobre%20las%20t%C3%A9cnicas%20de%20recopilaci%C3%B3n%20de%20datos,...%203%20Entrevistas%20...%204%20Focus%20group%20>
- [48] J. E. Borjas García, “Validez y confiabilidad en la recolección y análisis de datos bajo un enfoque cualitativo”, *TRASCENDER, CONTABILIDAD Y GESTIÓN*, núm. 15, pp. 79–97, sep. 2020, doi: 10.36791/tcg.v0i15.90.
- [49] R. R. Suarez, “El paradigma Positivista en la investigación”, *Departamento de Física y Matemática Universidad Nacional Experimental “Francisco de Miranda”*, pp. 4–6, jul. 2010, Consultado: el 9 de abril de 2023. [En línea]. Disponible en:

https://www.academia.edu/1668727/El_paradigma_Positivista_en_la_investigaci%C3%B3n

- [50] J. J. Espinosa Zúñiga, “Aplicación de metodología CRISP-DM para segmentación geográfica de una base de datos pública”, *Ingeniería Investigación y Tecnología*, vol. 21, núm. 1, pp. 1–13, ene. 2020, doi: 10.22201/fi.25940732e.2020.21n1.008.

Anexos

1. En el siguiente enlace se puede identificar el código del desarrollo de la simulación del brazo robótico:

<https://drive.google.com/file/d/1pZhky- pPsfIbChjtTvHtpz9hS7X399/view?usp=sharing>

2. En el siguiente enlace se visualizan los conjuntos de datos, tanto tabulares como de video e imágenes.

Conjunto datos tabulares:

<https://drive.google.com/drive/folders/1lsVhxYKWYiEdgIQ4xHgN7wVGkDT6tHif?usp=sharing>

Conjunto datos videos e imágenes:

<https://drive.google.com/drive/folders/10GlgWpZOuaYDIJ6ntKr0h3ig6lkvRMMY?usp=sharing>

3. En los siguientes enlaces se puede ingresar a los Script utilizados para realizar los modelos para experimentar con el aprendizaje en los conjuntos de datos tabulares y de video.

Aprendizaje con datos tabulares:

https://colab.research.google.com/drive/1OHFXAoAxM5WXTnV9yWZvU_eMLSswXurQZ?usp=sharing

Aprendizaje con datos videos e imágenes:

https://colab.research.google.com/drive/1OHFXAoAxM5WXTnV9yWZvU_eMLSswXurQZ?usp=sharing

 <p>UNIVERSIDAD CESMAG NIT: 800.109.387-7 VIGILADA MINEDUCACIÓN</p>	CARTA DE ENTREGA TRABAJO DE GRADO O TRABAJO DE APLICACIÓN – ASESOR(A)	CÓDIGO: AAC-BL-FR-032
		VERSIÓN: 1
		FECHA: 09/JUN/2022

San Juan de Pasto, 23 / 05 / 2024

Biblioteca
REMIGIO FIORE FORTEZZA OFM. CAP.
Universidad CESMAG
Pasto

Saludo de paz y bien.

Por medio de la presente se hace entrega del Trabajo de Grado / Trabajo de Aplicación denominado Aprendizaje de la cinemática de un brazo robótico para la simulación del movimiento mediante técnicas de Machine Learning, presentado por el (los) autor(es) Juan Angel Gualguan Oviedo, y Jhonatan Esteban Villarreal López del Programa Académico Ingeniería de sistemas al correo electrónico biblioteca.trabajosdegrado@unicesmag.edu.co. Manifiesto como asesor(a), que su contenido, resumen, anexos y formato PDF cumple con las especificaciones de calidad, guía de presentación de Trabajos de Grado o de Aplicación, establecidos por la Universidad CESMAG, por lo tanto, se solicita el paz y salvo respectivo.

Atentamente,



HÉCTOR ANDRÉS MORA PAZ
c.c. 1085251119
Ingeniería de sistemas
3172537641
hamora@unicesmag.edu.co.

 UNIVERSIDAD CESMAG <small>NIT: 800.109.387-7 VIGILADA MINEDUCACIÓN</small>	AUTORIZACIÓN PARA PUBLICACIÓN DE TRABAJOS DE GRADO O TRABAJOS DE APLICACIÓN EN REPOSITORIO INSTITUCIONAL	CÓDIGO: AAC-BL-FR-031
		VERSIÓN: 1
		FECHA: 09/JUN/2022

INFORMACIÓN DEL (LOS) AUTOR(ES)	
Nombres y apellidos del autor: Juan angel gualguan Oviedo	Documento de identidad: 1010137519
Correo electrónico: Jagualguan.7519@unicesmag.edu.co	Número de contacto: 3186504192
Nombres y apellidos del autor: Jhonatan esteban Villarreal López	Documento de identidad: 1004235758
Correo electrónico: Jhonatan.v.lopez.357@gmail.com	Número de contacto: 3157123698
Título del trabajo de grado: Aprendizaje de la cinemática de un brazo robótico para la simulación del movimiento mediante técnicas de Machine Learning	
Facultad y Programa Académico: Facultad: ingeniería Programa: ingeniería de sistemas	

En mi (nuestra) calidad de autor(es) y/o titular (es) del derecho de autor del Trabajo de Grado o de Aplicación señalado en el encabezado, confiero (conferimos) a la Universidad CESMAG una licencia no exclusiva, limitada y gratuita, para la inclusión del trabajo de grado en el repositorio institucional. Por consiguiente, el alcance de la licencia que se otorga a través del presente documento, abarca las siguientes características:

- a) La autorización se otorga desde la fecha de suscripción del presente documento y durante todo el término en el que el (los) firmante(s) del presente documento conserve (mos) la titularidad de los derechos patrimoniales de autor. En el evento en el que deje (mos) de tener la titularidad de los derechos patrimoniales sobre el Trabajo de Grado o de Aplicación, me (nos) comprometo (comprometemos) a informar de manera inmediata sobre dicha situación a la Universidad CESMAG. Por consiguiente, hasta que no exista comunicación escrita de mi(nuestra) parte informando sobre dicha situación, la Universidad CESMAG se encontrará debidamente habilitada para continuar con la publicación del Trabajo de Grado o de Aplicación dentro del repositorio institucional. Conozco(conocemos) que esta autorización podrá revocarse en cualquier momento, siempre y cuando se eleve la solicitud por escrito para dicho fin ante la Universidad CESMAG. En estos eventos, la Universidad CESMAG cuenta con el plazo de un mes después de recibida la petición, para desmarcar la visualización del Trabajo de Grado o de Aplicación del repositorio institucional.

 UNIVERSIDAD CESMAG <small>NIT: 800.109.387-7 VIGILADA MINEDUCACIÓN</small>	AUTORIZACIÓN PARA PUBLICACIÓN DE TRABAJOS DE GRADO O TRABAJOS DE APLICACIÓN EN REPOSITORIO INSTITUCIONAL	CÓDIGO: AAC-BL-FR-031
		VERSIÓN: 1
		FECHA: 09/JUN/2022

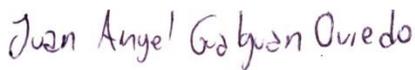
- b) Se autoriza a la Universidad CESMAG para publicar el Trabajo de Grado o de Aplicación en formato digital y teniendo en cuenta que uno de los medios de publicación del repositorio institucional es el internet, acepto(amos) que el Trabajo de Grado o de Aplicación circulará con un alcance mundial.
- c) Acepto (aceptamos) que la autorización que se otorga a través del presente documento se realiza a título gratuito, por lo tanto, renuncio(amos) a recibir emolumento alguno por la publicación, distribución, comunicación pública y/o cualquier otro uso que se haga en los términos de la presente autorización y de la licencia o programa a través del cual sea publicado el Trabajo de grado o de Aplicación.
- d) Manifiesto (manifestamos) que el Trabajo de Grado o de Aplicación es original realizado sin violar o usurpar derechos de autor de terceros y que ostento(amos) los derechos patrimoniales de autor sobre la misma. Por consiguiente, asumo(asumimos) toda la responsabilidad sobre su contenido ante la Universidad CESMAG y frente a terceros, manteniéndose indemne de cualquier reclamación que surja en virtud de la misma. En todo caso, la Universidad CESMAG se compromete a indicar siempre la autoría del escrito incluyendo nombre de(los) autor(es) y la fecha de publicación.
- e) Autorizo(autorizamos) a la Universidad CESMAG para incluir el Trabajo de Grado o de Aplicación en los índices y buscadores que se estimen necesarios para promover su difusión. Así mismo autorizo (autorizamos) a la Universidad CESMAG para que pueda convertir el documento a cualquier medio o formato para propósitos de preservación digital.

NOTA: En los eventos en los que el trabajo de grado o de aplicación haya sido trabajado con el apoyo o patrocinio de una agencia, organización o cualquier otra entidad diferente a la Universidad CESMAG. Como autor(es) garantizo(amos) que he(hemos) cumplido con los derechos y obligaciones asumidos con dicha entidad y como consecuencia de ello dejo(dejamos) constancia que la autorización que se concede a través del presente escrito no interfiere ni transgrede derechos de terceros.

Como consecuencia de lo anterior, autorizo(autorizamos) la publicación, difusión, consulta y uso del Trabajo de Grado o de Aplicación por parte de la Universidad CESMAG y sus usuarios así:

- Permiso(permitimos) que mi(nuestro) Trabajo de Grado o de Aplicación haga parte del catálogo de colección del repositorio digital de la Universidad CESMAG por lo tanto, su contenido será de acceso abierto donde podrá ser consultado, descargado y compartido con otras personas, siempre que se reconozca su autoría o reconocimiento con fines no comerciales.

En señal de conformidad, se suscribe este documento en San Juan de Pasto a los 22 días del mes de 05 del año 2024.

	
Nombre del autor: jhonatan esteban Villarreal López	Nombre del autor: juan angel gualguan Oviedo
 <hr/> Nombre del asesor: Héctor Andrés mora	