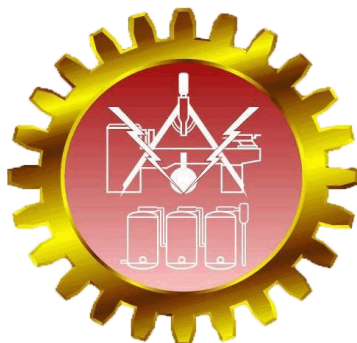


INSTITUTO TECNOLÓGICO DE AGUASCALIENTES



Tema:

Tesis para obtener el grado de:
Doctor en Ciencias de la Ingeniería

“MÁQUINA DE APRENDIZAJE EN PARALELO PARA OPTIMIZAR EL TIEMPO DE PROCESAMIENTO EN LA DETECCIÓN DE PATRONES RELACIONADOS CON EL SUICIDIO EN REDES SOCIALES”

PRESENTA:
MCI. Damian Martínez Díaz

Director:
Dr. Francisco Javier Luna Rosas
Codirector:
Dr. Julio César Martínez Romo

Aguascalientes, Aguascalientes, Diciembre 2024





Instituto Tecnológico de Aguascalientes
División de Estudios de Posgrado e Investigación

Aguascalientes, Ags., 09/enero/2025

No. De Oficio: 007/2025

Asunto: Autorización de Tesis de Grado Para su Impresión

M.C. DAMIAN MARTINEZ DIAZ
PRESENTE

Los abajo firmantes, Miembros del Jurado para Examen de Grado de Doctorado, hacemos CONSTAR que, habiendo revisado el Trabajo de Tesis desarrollado por Usted, bajo el título: **"MÁQUINA DE APRENDIZAJE EN PARALELO PARA OPTIMIZAR EL TIEMPO DE PROCESAMIENTO EN LA DETECCIÓN DE PATRONES RELACIONADOS CON EL SUICIDIO EN REDES SOCIALES"** hemos dictaminado que éste es de aceptarse, por lo que se autoriza su impresión.

ATENTAMENTE

*Excelencia en Educación Tecnológica
Ingenio, Cultura y Saber, que conducen a la Excelencia*

PRESIDENTE

DR. FRANCISCO JAVIER LUNA ROSAS

SECRETARIO

DR. JULIO CESAR MARTINEZ ROMO

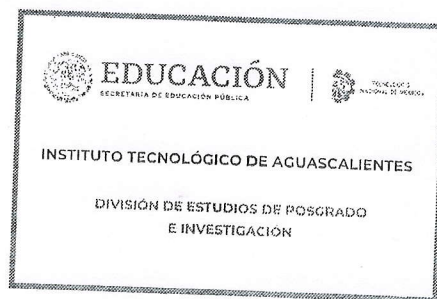
VOCAL

mar d 81
DR. MARIO ALBERTO RODRÍGUEZ DÍAZ

VOCAL

DR. MARCO ANTONIO HERNÁNDEZ VARGAS

VOCAL

DR. IVAN CASTILLO ZÚÑIGA

Av. Adolfo López Mateos #1801. Fracc. Bona Gens. C.P. 20256. Aguascalientes, Aguascalientes.
Tel. 449 910 5002 ext. 127; e-mail: jefatura.depi@aguascalientes.tecnm.mx
tecnm.mx | aguascalientes.tecnm.mx



2024
Felipe Carrillo
PUERTO
GOBERNADOR DEL GOBIERNO DEL ESTADO DE AGUASCALIENTES



Instituto Tecnológico de Aguascalientes
División de Estudios de Posgrado e Investigación

Aguascalientes, Ags., 09/enero/2025
Asunto: Autorización de uso de derechos

DAMIAN MARTINEZ DIAZ, mayor de edad, con domicilio ubicado en Manantial del sol # 605, Fraccionamiento Residencial Cedros, en mi calidad de titular y autor de la tesis denominada **"MÁQUINA DE APRENDIZAJE EN PARALELO PARA OPTIMIZAR EL TIEMPO DE PROCESAMIENTO EN LA DETECCIÓN DE PATRONES RELACIONADOS CON EL SUICIDIO EN REDES SOCIALES"** quien para todos los fines del presente documento se denominará **EL AUTOR Y/O TITULAR**, suscribo el presente documento de autorización de uso de derechos patrimoniales de autor a favor del Tecnológico Nacional de México/Instituto Tecnológico de Aguascalientes el cual se registrará por las cláusulas siguientes:

PRIMERA – AUTORIZACIÓN: **EL AUTOR Y/O TITULAR**, mediante el presente documento autoriza la utilización de los derechos patrimoniales de autor al Tecnológico Nacional de México/Instituto Tecnológico de Aguascalientes, de la tesis denominada **"MÁQUINA DE APRENDIZAJE EN PARALELO PARA OPTIMIZAR EL TIEMPO DE PROCESAMIENTO EN LA DETECCIÓN DE PATRONES RELACIONADOS CON EL SUICIDIO EN REDES SOCIALES"**, a través del Repositorio Institucional del Tecnológico Nacional de México (en lo sucesivo TecNM) y en el Repositorio Nacional, que puede ser consultado en la liga electrónica: (www.repositorionacionalcti.mx/).

SEGUNDA – OBJETO: Por medio del presente escrito, **EL AUTOR Y/O TITULAR** autoriza al Tecnológico Nacional de México/Instituto Tecnológico de Aguascalientes, a través del Repositorio Institucional del TecNM y en el Repositorio Nacional para que de conformidad con la Ley Federal del Derecho de Autor y la Ley de la Propiedad Industrial, use los derechos del documento antes referido, con fines exclusivamente académicos.

TERCERA – TERRITORIO: Los derechos aquí autorizados se dan sin limitación geográfica o territorial alguna.

CUARTA – ALCANCE: La presente autorización se da tanto para formato o soporte material, y se extiende a la utilización en medio óptico, magnético, electrónico, en red, mensajes de datos o similar conocido o por conocer, del ejemplar o número respectivo de la publicación.

QUINTA – EXCLUSIVIDAD: La autorización de uso aquí establecida no implica exclusividad en favor del Tecnológico Nacional de México/Instituto Tecnológico de Aguascalientes. Por lo tanto, **EL AUTOR Y/O TITULAR** en su carácter de autor de la obra objeto del presente documento se reserva el derecho de publicar directamente, u otorgar a cualquier tercero, autorizaciones de uso similares o en los mismos términos aquí acordados.





SEXTA – DERECHOS MORALES (Créditos y mención): La Autorización de los derechos antes mencionados no implica la cesión de los derechos morales sobre los mismos por cuanto en conformidad con lo establecido en los artículos 18, 19, 20, 21, 22 y 23 de la Ley Federal de Derechos de Autor, dada la cuenta que estos derechos son inalienables, imprescriptibles, irrenunciables e inembargables. Por lo tanto, los mencionados derechos seguirán radicados en cabeza de EL AUTOR Y/O TITULAR, y siempre deberá mencionarse su nombre cuando se utilice la obra.

SÉPTIMA – AUTORÍA: EL AUTOR Y/O TITULAR, declara y ratifica que el material objeto de la presente fue realizada por él o por ella sin violar o usurpar derechos de Propiedad Intelectual de terceros.

ATENTAMENTE

Excelencia en Educación Tecnológica®

Ingenio, Cultura y Saber, que conducen a la Excelencia

Vo.Bo.

DAMIAN MARTINEZ DIAZ
AUTOR DE LA TESIS

DR. FRANCISCO JAVIER LUNA ROSAS
DIRECTOR DE LA TESIS

ccp. Archivo





ÍNDICE GENERAL

Capítulo 1. INTRODUCCIÓN.....	7
1.1 Antecedentes y motivaciones.....	7
1.2 Delimitación del problema.....	8
1.2.1 Problema.....	8
1.3 Objetivos.....	8
1.3.1 Objetivo general	8
1.3.2 Objetivos particulares	9
1.4 Estado del arte.....	9
1.4.1 Suicidio.....	9
1.4.2 Redes Sociales	11
1.4.3 Recolección de Datos	12
1.5 Justificación de la investigación	13
1.6 Conclusiones del capítulo	13
Capítulo 2. SUICIDIO.....	14
2.1 Definición etimológica del suicidio.....	14
2.2 Historia del Suicidio	14
2.3 Estadísticas Globales de Suicidio	16
2.4 Estadísticas generales de suicidio en México	17
2.5 Suicidios según causa que motivo el acto.....	19
2.6 Suicidio según lugar de ocurrencia en México.....	20
2.7 Suicidio según el medio empleado en México	22
2.8 Suicidios según entidad federativa.....	23
2.9 Ideación suicida y autolesiones.....	24
2.10 Factores de riesgo asociados al suicidio.....	26
2.11 Como prevenir el Acto Suicida	26
2.12 Conclusiones del capítulo.....	27
Capítulo 3. MACHINE LEARNING	28
3.1 Antecedentes de Machine Learning.....	28
3.2 Aprendizaje Supervisado	31
3.2.1 Regresión Lineal.....	31
3.2.2 Regresión Logística	32
3.2.3 Máquinas de Vectores de Soporte (SVM).....	34
3.2.4 K-Nearest Neighbors (K-NN)	37
3.2.5 Multi-layer Perceptron (MLP).....	39
3.2.6 Random Forest.....	44
3.3 Aprendizaje No Supervisado	48
3.3.1 K-Means	48

3.3.2	Agrupamiento Jerárquico	50
3.3.3	PCA (Análisis de Componentes Principales)	52
3.3.4	t-SNE (t-distributed Stochastic Neighbor Embedding)	55
3.3.5	LDA (Latent Dirichlet Allocation)	57
3.4	Aprendizaje por Refuerzo	59
3.4.1	Q-Learning.....	59
3.4.2	DDPG (Deep Deterministic Policy Gradient)	60
3.4.3	PPO (Proximal Policy Optimization)	62
3.5	Aplicaciones del aprendizaje automático (ML)	64
3.6	Conclusiones del capítulo	65
Capítulo 4.	ALGORITMOS BIOINSPIRADOS	66
4.1	Inteligencia de Enjambre	66
4.1.1	Optimización por enjambre de partículas	66
4.1.2	Optimización de colonia de hormigas	69
4.2	Algoritmos Evolutivos	70
4.2.1	Algoritmos Genéticos	71
4.3	Conclusiones del capítulo	76
Capítulo 5.	MATERIALES Y MÉTODOS.....	77
5.1	Métodos experimentales	77
5.2	Metodología propuesta	80
5.2.1	Formulación del problema.....	80
5.2.2	Sistema de Computo	80
5.2.3	Colección de Datos	81
5.2.4	Preprocesamiento de Texto	82
5.2.5	Extracción y agrupación de características.....	83
5.2.6	Generación del Modelo de Aprendizaje Maquina.....	84
5.2.7	Rendimiento en el tiempo del Modelo	85
5.2.8	Métricas y métodos de evaluación de resultados	86
5.3	Materiales.....	90
5.4	Conclusiones del capítulo	92
Capítulo 6.	RESULTADOS Y DISCUSIÓN	93
6.1	Planificación de tareas	93
6.2	Técnica de Ventana deslizante	93
6.3	Experimento de Planificación de Tareas con AG	94
6.4	Experimento de Planificación de Tareas con PSO	100
6.5	Experimento de Planificación de Tareas con ACO	106
6.6	Precisión de predicción en clasificadores	112
6.7	Optimización de tiempo	116
6.8	Conclusiones del capítulo	122
CONCLUSIONES GENERALES		124
REFERENCIAS		125

ÍNDICE DE FIGURAS

Figura 2.1 Tasa de Suicidios en la población de 15 a 29 años [50].	17
Figura 2.2 Agrupación de las Estadísticas de Suicidios en México desde 1970-2021.	18
Figura 2.3 Registro de los Intentos de Suicidio 1980 – 2005.	19
Figura 2.4 Ocurrencia y sexo según causa que motivó el acto [52].	20
Figura 2.5 Suicidio según lugar de ocurrencia en México 2006 – 2011 [55].	21
Figura 2.6 Medio utilizado por hombres y mujeres 2010-2021 [56].	23
Figura 2.7 Suicidios según entidad federativa 2010-2021 [57].	24
Figura 2.8 Población de 10 años y más que ha pensado suicidarse [58].	25
Figura 2.9 Población de 10 años y más que ha pensado suicidarse por sexo y frecuencia de ideación suicida [58].	25
Figura 3.1 Hiperplanos de separación en un espacio bidimensional de un conjunto de ejemplos separables en dos clases [76].	34
Figura 3.2 Descripción del algoritmo KNN [77].	39
Figura 3.3 KNN clasificador con k pequeño, mediano y grande.	39
Figura 3.4 Capa Oculta de Algoritmo MLP	40
Figura 3.5 Bosques aleatorios (Random Forest)	45
Figura 3.6 Medida de Inercia en Algoritmo K-Medias [73].	49
Figura 3.7 Fusión jerárquica de conglomerados [73].	51
Figura 3.8 Análisis de componentes principales [73].	53
Figura 3.9 t-SNE incrustación de los dígitos [83].	55
Figura 3.10 LDA Modelo generativo de tres niveles [83].	57
Figura 3.11 Tabla Q Acciones – Estados.	60
Figura 3.12 Algoritmo DDPG [86]	62
Figura 3.13 Estilo actor-crítico del algoritmo PPO [87].	63
Figura 4.1 Diagrama General del PSO.	67
Figura 4.2 Actualización de la posición de la i-th partícula.	68
Figura 4.3 Hormigas generando Soluciones de nodo origen a nodo destino.	70
Figura 4.4 Diagrama General de Algoritmo Genético.	72
Figura 5.1 Metodología propuesta.	80
Figura 5.2 Técnica Tabla Testing ó Training Holdout	89
Figura 5.3 Técnica K-Fold.	90
Figura 5.4 Técnica Leave One Out.	90
Figura 6.1 Ventana Deslizante.	93
Figura 6.2 Permutación de cromosomas.	94
Figura 6.3 Fitness por cada uno de los cromosomas.	97
Figura 6.4 Selección de dos cromosomas con tecnica de ruleta.	98
Figura 6.5 Cruzamiento de cromosomas.	99
Figura 6.6 Mutación de cromosomas.	99

Figura 6.7 Permutación de partículas.	101
Figura 6.8 Posición de partículas.....	105
Figura 6.9 Permutación de hormigas.....	107
Figura 6.10 Caminos de las hormigas.	110
Figura 6.11 Comparación de número de vectores en Word2vec.....	115
Figura 6.12 Comparación secuencial vs AG.	117
Figura 6.13 Comparación secuencial vs PSO.....	118
Figura 6.14 Comparación secuencial vs ACO.	119
Figura 6.15 Comparación Bioinspirada.....	120
Figura 6.16 Comparación Algoritmos tradicionales vs Algoritmos Bioinspirados.....	120
Figura 6.17 Rendimiento de los Procesadores con estrategia bioinspirada.....	122
Figura 6.18 Rendimiento de los Procesadores en Procesamiento Secuencial.....	122

ÍNDICE DE TABLAS

Tabla 2.1 Ocurrencia y sexo según causa que motivó el acto [52].	19
Tabla 2.2 Suicidio según lugar de ocurrencia en México [54].	21
Tabla 2.3 Medio empleado para el suicidio 2010-2021 [56].	22
Tabla 3.1 Funciones Kernel del SVM	35
Tabla 4.1 Tupla de Padres seleccionados.	74
Tabla 4.2 Descendiente del padre X.	75
Tabla 4.3 Descendiente del padre Y.	75
Tabla 4.4 Descendencia de los padres X y Y.	75
Tabla 4.5 Mutación del Descendiente X.	75
Tabla 5.1 Vocabulario del suicidio en inglés y español.	81
Tabla 6.1 Estado del Actual de los Procesadores.	94
Tabla 6.2 Tareas asignadas en los procesadores.	95
Tabla 6.3 Probabilidad acumulada por cromosoma	97
Tabla 6.4 Nuevo contenido de Ventana Deslizante.	100
Tabla 6.5 Estado del Actual de los Procesadores.	101
Tabla 6.6 Tareas asignadas en los procesadores.	102
Tabla 6.7 Posición y velocidad por partícula.	104
Tabla 6.8 Nuevo contenido de Ventana Deslizante.	106
Tabla 6.9 Estado del Actual de los Procesadores.	107
Tabla 6.10 Tareas asignadas en los procesadores.	108
Tabla 6.11 Visibilidad y Feromonas por hormiga.	109
Tabla 6.12 Actualización de la probabilidad de la hormiga.	111
Tabla 6.13 Nuevo contenido de Ventana Deslizante.	112
Tabla 6.14 Conjunto de Datos preprocesado con Modelo propuesto	113
Tabla 6.15 Validación del Modelo con LDA	113
Tabla 6.16 Algoritmos Experimentales y Técnicas de Validación	115
Tabla 6.17 Calibración de GA	116
Tabla 6.18 Calibración del PSO	117
Tabla 6.19 Calibración del ACO	118
Tabla 6.20 Mejora de Tiempo Secuencial vs Bioinspirado.	120

AGRADECIMIENTOS

En primer lugar quiero agradecer a Ángeles por su apoyo incondicional en este gran proyecto de mi vida, a mis hijos Iñaki y Kitzia, cuyo apoyo y paciencia que fue puesta a prueba en incontables ocasiones, a mis padres, por darme la vida y siempre creer en mí.

Me gustaría agradecer al director de tesis Dr. Francisco Javier Luna Rosas por su guía experta y apoyo continuo durante todo el proceso de investigación. Su experiencia y dedicación contribuyeron en gran medida al diseño y mejora de este trabajo.

También me gustaría expresar mi más sincero agradecimiento a todos aquellos que, de una forma u otra, contribuyeron a la realización de esta tesis doctoral. Este logro no hubiera sido posible sin el legado de quienes han contribuido a este campo de estudio.

Agradezco al TecNM / Instituto Tecnológico de Aguascalientes por todo el soporte institucional brindado para la realización y conclusión de este proyecto doctoral.

Capítulo 1. INTRODUCCIÓN

Este capítulo presenta el protocolo de investigación, proporcionando al lector los principales antecedentes, problemáticas y motivaciones relacionados al suicidio que impulsaron la realización de este proyecto. Su objetivo es facilitar la comprensión del análisis de resultados y las conclusiones, estableciendo las bases del trabajo a desarrollar y los parámetros para la interpretación de los hallazgos.

1.1 Antecedentes y motivaciones

La prevención del suicidio sigue siendo un desafío universal. Cada año, el suicidio se encuentra entre las 20 principales causas de muerte a nivel mundial para personas de todas las edades y como muestra significativa es el tercer lugar entre jóvenes de 15 a 19 años. Es responsable de más de 700.000 muertes, lo que equivale a un suicidio cada 40 segundos [1]. Cada vida perdida representa al compañero, hijo, padre, amigo o colega de alguien, esto equivale a 108 millones de personas al año que están profundamente afectadas por la conducta suicida. La conducta suicida incluye el suicidio e ideación suicida y los intentos, ya que, por cada suicidio, 25 personas hacen un intento de suicidio y muchos más tienen ideación suicida. La conducta suicida es el resultado de una convergencia de factores de riesgo tanto genéticos, psicológicos, sociales y culturales, etc., a veces combinados con experiencias de trauma y pérdida [2]. Desde el 2003, la Asociación Internacional para la Prevención del Suicidio, en colaboración con la Organización Mundial de la Salud (OMS), ha promovido cada 10 de septiembre el Día Mundial para la Prevención del Suicidio. El objetivo de este día es concientizar a nivel mundial que el suicidio puede prevenirse [3]. De acuerdo con los datos publicados en 2021, por el Instituto Nacional de Estadística y Geografía (INEGI) [4], en México, se tuvieron más de 80 mil defunciones por causas externas, de las cuales el 9.96% equivale a 8,432 muertes clasificadas como presunto suicidio, de estos 1,568 (18.6%) corresponden a mujeres y 6,850 (81.2%) a hombres entre grupos de edades de 15 a 24 y de 25 a 34 años con un 24.5% y 27.4% respectivamente. Estas lesiones auto infligidas intencionalmente representan una tasa de incremento por cada 100 mil habitantes de 6.2% a 6.6% en este tipo de desorden en referencia al año inmediato anterior. Las tasas de suicidios varían entre zonas geográficas y entidades federativas mexicanas, de las cuales las entidades con una mayor tasa son Chihuahua con 15.1%, Yucatán con 14.5% y Aguascalientes con 12.0%. Acorde a los datos preliminares para 2023 [5] no son nada alentadores, lo cual sumado con el propósito de reducir los índices de suicidio se fomentan en todo el mundo compromisos y medidas prácticas para

prevenirlo; la tasa de suicidios sigue en aumento. El suicidio es considerado un problema complejo que se ha analizado desde diversas perspectivas tal como psicosocial, biológica, psicológica, antropológica, sociológica, económica y geográfica. Por ello es de vital importancia la generación de modelos predictivos en miras de prevenir el suicidio a partir del análisis del contexto y las circunstancias sociales actuales que han inferido en la decisión de los usuarios en las redes sociales en una perspectiva multifactorial [6]. El termino multifactorial refiere a los diversos factores que intervienen en el suicidio tales como la gran cantidad de cambios sociales, en el sector familiar, económico, laboral, escolar, salud y la violencia.

1.2 Delimitación del problema

Se diseñará el modelo teniendo como escenario datos de la red social Twitter, específicamente orientado al análisis de contenido relacionado con el suicidio, que ha sido publicado por los usuarios delimitando dicho contenido en solo al idioma inglés. El modelo estará limitado al uso de algoritmos de máquinas de aprendizaje de clasificación supervisada.

1.2.1 Problema

El suicidio es un fenómeno que se ha manifestado a lo largo de la historia entre hombres y mujeres de todas las edades [7] y en todas las sociedades y, por tanto, ha sido interpretado de forma diferente según la sociedad y la época en la que se produjo. Solo en México representa la segunda causa de muerte entre jóvenes de 15 a 29 años, y recibe toda la atención de las autoridades de salud pública ya que su incidencia ha aumentado rápidamente en los últimos años en paralelo con la difusión y el uso de las redes sociales de internet. Es necesario que los padres y familiares de quienes se encuentran en dicho rango de edad, así como los docentes, los profesionistas de diversas disciplinas y la sociedad en general, se preparen y sumen esfuerzos para frenar el aumento de este problema. Haciendo esto un desafío, el generar modelos con la capacidad de ayudar a prevenir los suicidios en las redes sociales, dando la oportunidad de que amigos o seguidores de los usuarios con estas tendencias, puedan ser apoyados, buscando reducir el número de muertes que cada año se incrementa en nuestro país y a nivel global.

1.3 Objetivos

1.3.1 Objetivo general

Diseñar un modelo para optimizar el tiempo de procesamiento en la detección de patrones relacionados con el suicidio en Redes Sociales.

1.3.2 Objetivos particulares

- Creación de una base de datos de la red social Twitter relacionada con el suicidio.
- Aplicar técnicas de preprocesamiento de datos para limpiar los tweets obtenidos de redes sociales.
- Aplicar técnicas de procesamiento del lenguaje natural (PNL) para la extraer patrones relacionados con el suicidio.
- Diseñar un modelo de aprendizaje automático para identificar patrones asociados con el suicidio.
- Optimización del modelo para reducir los tiempos en la extracción de patrones relacionados con el suicidio.

1.4 Estado del arte

1.4.1 Suicidio

El suicidio es un problema de salud mundial prevenible, cuyo origen puede estar influenciado por factores psicológicos, sociales y recientes acontecimientos globales como la pandemia de COVID-19 [8] [9] [10]. Entre las causas principales se incluyen trastornos psicológicos, presión social, traumas, antecedentes familiares, consumo de sustancias y aislamiento social, entre otros. Según la OMS, en 2019, el suicidio fue responsable de más de 703,000 muertes, superando otras causas importantes como el VIH/SIDA o el cáncer de mama [1]. Por ello, la OMS lo ha priorizado dentro de los Objetivos de Desarrollo Sostenible, impulsando estrategias integrales de prevención, mejoras en los sistemas de información y avances en la investigación sobre salud mental, con una meta fijada para 2030 [11].

La problemática del suicidio afecta globalmente a los hombres el doble que a las mujeres, con tasas de 12.6% por 100,000 hombres frente a 5.4% por 100,000 mujeres. Los hombres presentan tasas más altas en países de ingresos altos (16.5% por 100,000), mientras que las mujeres tienen tasas más elevadas en países de ingresos medianos y bajos (7.1% por 100,000). Entre 2000 y 2019, las tasas de suicidio a nivel mundial disminuyeron un 36%, con caídas que varían del 17% en la Región del Mediterráneo Oriental al 49% en el Pacífico Occidental. Sin embargo, en la Región de las Américas, las tasas aumentaron un 17% durante el mismo periodo [12].

En México, en 2020, ocurrieron 82,449 defunciones por causas externas, de las cuales el 9.57% equivale a 7,896 muertes clasificadas como presunto suicidio, de estos 1,439 (18.2%) corresponden a mujeres y 6,449 (81.7%) a hombres, no especificándose sexo para 8 casos. Los grupos más afectados fueron jóvenes de 15 a 24 años y adultos de 25 a 34 años, concentrando la mayor proporción de suicidios con un 25.6% y 25.9% respectivamente. Las principales causas fueron ahorcamiento (84.4%) y armas de fuego (6.7%). La tasa nacional fue de 6.2% suicidios por cada 100,000 habitantes, destacando Chihuahua (14.4%), Aguascalientes (12.6%) y Yucatán (10.45%) con las tasas más altas, mientras que Guerrero (1.4%), Baja California (2.6%) y Veracruz (3.6%) tuvieron las más bajas [13].

La investigación sobre el "texto suicida" se ha centrado en las notas de suicidio [14], consideradas una expresión prototípica de quienes sufren. Este fenómeno ha sido estudiado desde perspectivas psicológicas y psiquiátricas [15], dado que las personas en crisis suelen experimentar un profundo sufrimiento en silencio, lo que dificulta prevenir intentos de suicidio. Aunque existen marcadores biológicos y psicológicos, la detección de factores de riesgo se realiza mediante herramientas clínicas estándar que ayudan a comprender las razones detrás de estas decisiones fatales y a prevenir futuros suicidios [16] [17] [18] [19]. Desde el ámbito de las ciencias de la computación, se han desarrollado métodos para identificar patrones suicidas utilizando datos no clínicos [20] [21] [22] [23]. Se han creado modelos automáticos de clasificación de textos escritos y orales en internet, considerando que la población global es de más 7,91 millones de personas, de los cuales el 62.5% son usuarios activos de internet que equivale a 4.92 millones, 67.1% son usuarios de teléfonos móviles que equivalen a 5.31 millones y el 58.4% son usuarios de medios sociales que equivale a 4.62 millones [24], este último representa una poco más de la mitad de la población mundial donde se comparte contenido que se publica en las redes sociales como Facebook, Twitter, Instagram entre muchas otras.

La depresión, principal causa de discapacidad y suicidio, puede detectarse en el lenguaje usado en estas redes sociales [25]. Se emplean técnicas de procesamiento de lenguaje natural (PNL) y aprendizaje automático [25] [26] [27], se analizan publicaciones para identificar señales de riesgo, facilitando intervenciones tempranas en poblaciones vulnerables. Sin embargo, se han propuesto pocos enfoques automatizados para extraer estresores psiquiátricos de Twitter, se ha buscado reconocer estos estresores relacionados con el suicidio utilizando métodos basados en el aprendizaje profundo y la estrategia de transferencia de aprendizaje que aprovecha un conjunto de datos de anotación existente del texto clínico [28].

Una de las mejores formas de prevenir el suicidio es que las personas en apuros escuchen a las personas que se preocupan por ellos [7]. Facebook está en una posición única, a través de su apoyo a las redes y amistades en el sitio, para ayudar a conectar a una persona en estas situaciones difíciles con personas que puedan apoyarla. Las publicaciones relacionadas con el suicidio son significativamente más propensas a recibir una respuesta.

El tiempo promedio de respuesta a una publicación relacionada con el suicidio fue de poco más de 1 hora, aunque el tiempo medio de respuesta fue inferior a 5 min [29]. Esto fue significativamente más rápido que la respuesta al tiempo de publicaciones no relacionadas con el suicidio. Esto sugiere que aumentar el número de seguidores o amigos puede no conducir a una mayor probabilidad de respuesta, aunque este hallazgo necesita ser replicado en un estudio más amplio.

El sociólogo francés Émile Durkheim, en su influyente libro *Suicide* [36], sugiere que, aunque el suicidio es un acto individual con causas individuales, las tasas de suicidio son hechos sociales y pueden explicarse por dos dimensiones del entorno social: integración y regulación. La integración social se refiere al grado en que los miembros de un grupo social comparten sentimientos colectivos, y la regulación social es el grado en que los individuos están limitados por las normas y valores sociales. Los países asiáticos como China representan una gran cantidad de suicidios en todo el mundo, y la mayoría ocurren en áreas rurales [21]. Los trastornos por suicidio y consumo de alcohol son importantes disparidades de salud experimentadas por jóvenes y hombres en particular, con un riesgo significativamente mayor de muerte por problemas relacionados con el suicidio. Las plataformas de comunicación en línea se utilizan cada vez más para expresar pensamientos suicidas. Existe un interés considerable en monitorear tales mensajes, tanto para propósitos de prevención poblacional e individual, como para informar sobre investigación y políticas de suicidio. La sobrecarga de información en línea prohíbe la detección manual como alternativa [37] investiga la clasificación de texto supervisado para modelar y detectar el suicidio en los mensajes del foro.

1.4.2 Redes Sociales

Recientemente, ha habido algunos trabajos destacados de análisis de redes sociales, especialmente sobre el tema análisis de opinión de Twitter. La mayoría de estos intentos de extraer pensamientos y opiniones públicas se basan en técnicas de extracción de sentimientos. La red social es más que nunca un medio de comunicación para el intercambio de información. Ofrece una información considerable a gran velocidad y los servicios se adaptan más a las necesidades de los usuarios. Estos pueden consultar Internet para crear comunicaciones e interacciones sociales (entre individuos o grupos de individuos) y creación de contenido [38]. Durante los últimos años, Internet ha visto un alcance más amplio a través del desarrollo de las redes sociales. Basados en técnicas de comunicación fáciles y accesibles a todos, los medios promueven la interacción social a través de Internet. Al ofrecer acceso gratuito, las redes sociales han sido promovidas masivamente y generado un debate público en Internet. Existen aproximadamente 900 sitios de redes sociales disponibles en Internet [39]. La rapidez con que la información es difundida en una red social es increíble, por ejemplo el crimen asociado con el tiroteo

puede detectarse en Twitter en tan solo 10 minutos después de ocurrido el incidente, sin embargo, el informe de noticias típico tarda aproximadamente 3 horas [40].

Las redes sociales como Twitter y Facebook están cada vez más asociadas con fenómenos como el acoso, la intimidación, suicidio o incluso la depresión. Por lo tanto, es muy importante detectar las posibles víctimas lo antes posible para fortalecer la prevención de estos fenómenos en la web. De hecho, podemos citar como ejemplo el caso de dos raperos estadounidenses, Freddy E. y Capital Steez [20], a quienes se les da la muerte comentando en vivo sus acciones en sus cuentas de Twitter. Este tipo de suicidios de celebridades hacen que los usuarios de las redes sociales respondan a lo sucedido de una forma desfavorable que se ha identificado como un fenómeno "imitador", denominado el "Efecto Werther" [41]. Las publicaciones en Twitter están relacionadas con el cambio en las tasas de suicidio después de la muerte de una celebridad mucho más fuerte que el volumen de informes de los medios en los periódicos o en la televisión. Las tres categorías más eficientes de intervención como la limitación del acceso a medios letales, la preservación del contacto con los pacientes hospitalizados por un intento de suicidio después de la hospitalización y la implementación de emergencias, es información útil para la reflexión y el desarrollo de acciones para por la prevención del suicidio [42] .

Estas redes sociales han provisionado a la comunidad científica sus plataformas para poder extraer información de diferentes tópicos que ayuda a los investigadores a generar métodos automáticos que permitan identificar patrones como la depresión, los trastornos de estrés y el suicidio, con el uso algoritmos de aprendizaje automático o Machine Learning (ML) o Deep Learning (DL) [43]. Una de las redes sociales más utilizadas en el mundo es Twitter con un promedio de más de 58 millones de tweets generados por día, realizados en mensajes cortos de 280 caracteres donde expresan sus ideas, emociones, o inclusive noticias [40]. Con los mensajes que los usuarios comparten en esta red se han propuesto técnicas para detectar patrones, de depresión, estrés, suicidio entre otros desórdenes mentales, pero la mayoría de ellas no aprovechan el contenido del texto para reducir el impacto de la escasez de datos como la realizada por [44] donde propone un modelo multitarea, que realizan las tareas en conjunto de detección de depresión y clasificación de sentimiento como una tarea secundaria y como tarea principal el reconocimiento de la emoción de las frases de suicidio.

1.4.3 Recolección de Datos

Estos "tweet" son una publicación corta de mensajes en Twitter. Los tweets se crean en tiempo real, con la escasez garantizada en el límite de 280 caracteres de las publicaciones y la popularidad de las aplicaciones móviles, los usuarios tuitean y retuitean instantáneamente. Un ejemplo de la velocidad de la información es por ejemplo, el crimen asociado con el tiroteo puede detectarse en Twitter en tan solo 10 minutos después de ocurrido el incidente, sin embargo, el informe de noticias tarda aproximadamente 3 horas

[40]. Las redes sociales como Twitter y Facebook están cada vez más asociadas con fenómenos como el acoso, la intimidación, suicidio o incluso la depresión [20]. Por lo tanto, es muy importante detectar las posibles víctimas lo antes posible para fortalecer la prevención de estos fenómenos en la web.

1.5 Justificación de la investigación

El suicidio es un fenómeno humano universal que ha estado presente en todas las épocas históricas. Sin embargo, las distintas sociedades lo han materializado hacia las actitudes enormemente variables en función de los principios filosóficos, religiosos e intelectuales de cada cultura. La importancia de esta investigación se centra en buscar analizar uno de los fenómenos más complejos que enfrenta el ser humano como lo es suicidio y en particular la forma como se expresan los usuarios en redes sociales como Twitter. Conocer mejor este problema nos permitirá idear estrategias útiles para prevenir dicho comportamiento. Esta investigación, también puede generar hipótesis para futuras investigaciones y ser el inicio de una serie de estudios sobre la conducta suicida en redes sociales.

1.6 Conclusiones del capítulo

En este capítulo se analizó la problemática del suicidio como un problema de salud global y debe ser abordado por los sectores multidisciplinarios involucrados, y todos deben poner de su parte para lograr prevenir este tipo de eventos que afectan a varias personas alrededor del mundo. Crear un modelo automatizado para la prevención del suicidio es una necesidad para lograr la clasificación automática de opiniones con un esfuerzo multidisciplinario, donde la lingüística y las técnicas de PNL juegan un papel importante para intentar identificar los factores de riesgo entre los usuarios de las redes sociales, y de esta forma ofrecer una alternativa en la prevención del suicidio en tiempo real.

Capítulo 2. SUICIDIO

Este capítulo presenta el suicidio como un fenómeno que varía a la época y condiciones de un país. Diferentes ramas de la ciencia han estudiado este fenómeno como la psicología, sociología, psiquiatría, antropología, epidemiología y la demográfica, basado en la complejidad y el impacto que tiene en la mente humana, repercutiendo en diferentes ámbitos como el familiar, el económico, social y por supuesto la salud pública. Siempre buscando explicarlo, justificarlo o reprobarlo, analizarlo, clasificarlo y, claro, predecirlo con el objeto de hacer posible su prevención. Considerado al suicidio como una más de las enfermedades mentales a lo largo de tiempo, ya que no se declara la depresión que este asociado al acto de matarse, solo se hace referencia al acto de matarse por una mente alterada (locura, insanidad e ira).

2.1 Definición etimológica del suicidio

Etimológicamente, la palabra suicidio proviene del latín sui (sí mismo) y cidium (matar), que en conjunto significa "matarse a sí mismo", "atentando contra la propia vida", o "matar-se". Los griegos expresaban el suicidio como auto (auto) y keiria (mano), lo que se podría expresar como "muerte elegida por uno mismo" o "muerte ejecutada por mano propia".

2.2 Historia del Suicidio

En todos los tiempos, el hombre ha interpretado ejercer sobre sí mismo el derecho de vida y muerte, como lo detallado en el libro [45], que da paso a grandes periodos de la historia donde analiza por país y época como llevaron al hombre al suicidio. Iniciando con los países de Asia, en la India en específico desde la antigüedad, bajo la influencia del brahmanismo, los sabios de la época buscaban el nirvana, y estos se suicidaban durante las fiestas religiosas. En Grecia antigua es una de las primeras culturas que aborda seriamente el problema del suicidio, más allá de un análisis puramente filosófico y hasta político planteado por Aristóteles y Platón. En Atenas se castigaba el suicidio, privando al cadáver de una sepultura, su mano derecha era enterrada en otro lugar después de ser amputada,"...Además, deben enterrarlos sin fama en los confines de los doce distritos en aquellos lugares que sean baldíos y sin nombre, sin señalar sus tumbas con estelas o nombres" [46].

En Roma, se tuvieron dos etapas de reflexión, la primera con Cicerón (106-43) que avalaba el suicidio si era en una forma de heroísmo, amor, abnegación o defensa del honor,

la segunda fue con Constantino (290-377) que penalizo el suicidio tomando medidas estrictas como la confiscación de los bienes de las familias después del suicidio para compensar al estado por la pérdida de un ciudadano, ya en el periodo de la decadencia del imperio Romano, los suicidios se incrementaron, este era prohibido entre los esclavos y el que lograba sobrevivir después de atentar contra su vida se le castigaba severamente, los soldados al pertenecer al estado no podían disponer de sus vidas, pero podía autodestruirse en caso de sufrimiento intolerable. Galia y Germania, los suicidios religiosos fueron muchos, buscando que se abrieran ante ellos las puertas del Walhalla, en Germania, era frecuente el suicidio entre anciano y mujeres, las viudas se sacrificaban sobre los cuerpos de los esposos.

Los galos preferían la muerte antes de ser esclavos por los invasores romanos, los visigodos al envejecer se arrojaban desde una roca llamada la Roca de los ancestros, para ganarse con su valentía la entrada al paraíso. En la época del cristianismo en el tiempo de Constantino comenzó a castigar el suicidio, pero este castigo no divino de una Ley bíblica propiamente. Acorde a los diez mandamientos de Moisés hacían referencia en su sexta regla al “acto de Matar”, no al “acto de Matarse”. Para los intelectuales cristianos, el suicidio fue tan complejo que se necesitó hacer una reinterpretación de este mandamiento, ya que con el “martirio voluntario” la iglesia tomo el suicidio con preocupación, ya que muchos fieles cayeron en manos de los romanos y fueron asesinados, aunque los fieles lo vieron como una forma de sacrificio por la fe, logrando el beneficio divino o paraíso. San Agustín es un importante personaje en la historia del suicidio en la era católica, ya que busco una razón para considerar el suicidio como un pecado, encontrando la justificación al sexto mandamiento “Aplicaremos al hombre las palabras no mataras, entendiendo: ni a otro ni a ti, puesto que quien se mata a si mismo mata a un hombre” agregando “... sabemos que no existe ley alguna que permita quitar la vida, incluso al culpable, por iniciativa privada y, por tanto, quien se mata a si mismo es homicida” [47].

En México, acorde [48], se recolecta la información de periódicos que basan las noticias del siglo XIX a los inicios del siglo XX, basándose en las conductas impuestas por la iglesia católica, donde se ve el suicidio como una preocupación del tipo moral y religioso. A finales del siglo XIX una persona que se suicidaba no necesariamente debería estar deprimida, ya que las causas podrían ser variadas entre lo moral y lo religioso, la teoría registrada en esta época de la depresión como teoría fue adoptada en el siglo XX, pero ya se empezaba a detectar que había una relación cuando una persona se suicidaba, anímicamente algo estaba mal.

Hablar del suicidio no puede dejar de nombrarse a Émile Durkheim (1858-1917) quien es considerado uno de los padres de la sociología. En el libro *El Suicidio* [36], Durkheim analiza diversas variables y datos estadísticos para explicar las tasas de suicidio en varios países europeos. Identifica cuatro tipos de suicidios relacionados con estados mentales específicos [36]:

- Suicidio maniático: Resulta de consecuencia de alucinaciones o de concepciones delirantes, caracterizado por cambios rápidos en las emociones y motivos.
- Suicidio melancólico: Se relaciona con un estado general de extrema depresión, y exagerada tristeza, donde la persona pierde el interés por los placeres de la vida y busca el suicidio como una salida.
- Suicidio obsesivo: Se relaciona al estar obsesionado por el deseo de matarse sin ningún motivo racional para hacerlo.
- Suicidio impulsivo o automático: Ocurre debido a una impulsión repentina y abrumadora, sin un proceso de reflexión previo.

En otro estudio comparativo [49] también clasifica los suicidios en tres categorías basadas en la integración social.

- Suicidios egoístas: Resulta cuando hay una falta de integración social, ligado a una desconexión del entorno por el individualismo.
- Suicidio altruista: Se relaciona en sociedades rígidamente estructuradas con fuertes normas grupales, donde el individuo sacrifica su vida por el bien grupal.
- Suicidio anómico: Resulta en contextos de desregulación social, como crisis económicas o guerras, que generan desorientación individual y a un sentimiento de falta de significación en la vida.

Durkheim argumenta que el suicidio es un fenómeno social influenciado por factores externos, y su estudio revela correlaciones entre la situación social y las decisiones individuales relacionadas con el suicidio.

2.3 Estadísticas Globales de Suicidio

La prevención del suicidio es un desafío universal. Cada año, el suicidio se encuentra entre las 20 principales causas de muerte a nivel mundial para personas de todas las edades y como muestra significativa es el cuarto lugar entre jóvenes de 15 a 19 años por detrás de los traumatismos debidos al tránsito, la tuberculosis y la violencia interpersonal. Es responsable de más de 700.000 muertes, lo que equivale a un suicidio de cada 100 muertes en el mundo. El suicidio afecta a nivel mundial al doble a los hombres que, a las mujeres, con el 12.6% por 100,000 hombres, 5.4% por 100,000 mujeres. La tasa de suicidio de hombres sucede generalmente en países de ingresos altos con un 16.5% por 100,000 habitantes, en cambio la tasa de mujeres se presenta más comúnmente entre países de mediano bajo ingreso con un 7.1% por 100,000 habitantes. Las tasas de suicidio disminuyeron en los casi 20 años transcurridos entre 2000 y 2019; la tasa mundial se redujo en un 36%, con descensos que oscilaron del 17% de la Región del Mediterráneo Oriental al 47% de la Región de Europa y el 49% del Pacífico Occidental. En cambio, en la Región de las Américas, las tasas se incrementaron en un 17% en ese mismo periodo, el 77% de los suicidios se producen en los países bajos y medianos, destacando entre las

principales causas la ingesta de plaguicidas, ahorcamientos y disparó con arma de fuegos. [12].

2.4 Estadísticas generales de suicidio en México

En México, en 2021, se reportaron en el país un total de fallecimientos de 1,093.210 de los cuales 8,351 muertes fueron clasificadas como como lesiones autoinfligidas, lo cual representa una tasa de suicidio de 6.5% por cada 100,000 habitantes. de estos 1,552 (2.4%) corresponden a mujeres y 6,785 (10.9%) a hombres. Entre las diez principales causas de muerte se encuentra el suicidio donde los grupos de edades por lesiones autoinfligidas intencionalmente (suicidios) están entre los 10 a 14 años con un 2.1%, 15 a 29 años 10.4%, 30 a 59 años 8.3% y de 60 y más 4.9% por cada 100,000 habitantes. Siendo el grupo de mayor riesgo de fallecimiento las personas de 15 a 29 años, seguido del grupo de 30 a 59 años. En la Figura 2.1, los datos reportados por el Instituto Nacional de Estadística y Geografía (INEGI) [50], el suicidio entre el grupo de personas de 15 a 29 años aumento en los últimos años, a partir del 2015 se estimó la tasa del 8.1 muertes por cada 100,000 habitantes, pero en el reporte del 2021 los suicidios pasaron a 10.4 por cada 100,000 habitantes, viendo un incremento relevante en los hombres quien en el 2015 tenían una estimación de suicidios de 12.4 incrementando al 2021 a 16.2, en contrasté con las mujeres que en las mismas fechas estaban en 3.9 e incrementaron a 4.8, incrementando así 1 muerte más en comparación con las casi 4 de los hombres.

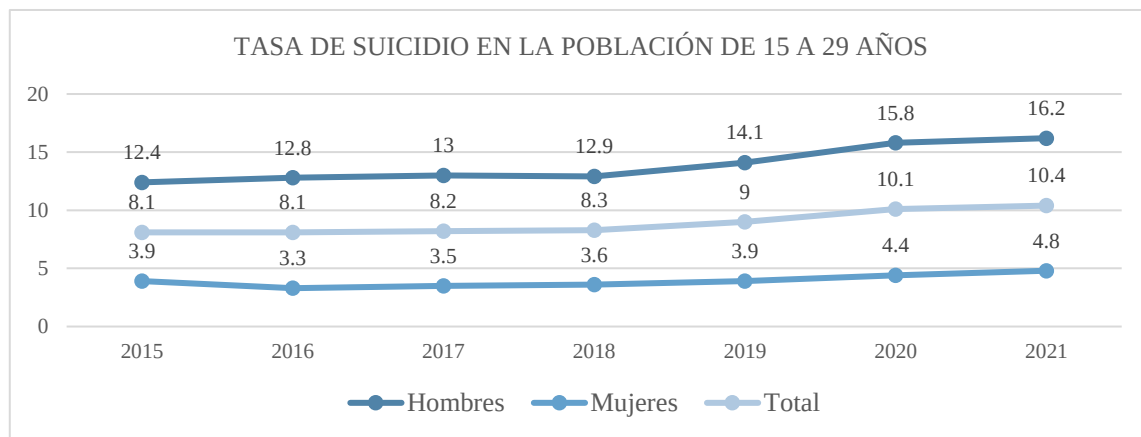


Figura 2.1 Tasa de Suicidios en la población de 15 a 29 años [50].

Los suicidios son diferenciados entre los homicidios, en base a las investigaciones que se realizan por las Agencias del Ministerio Público y que con frecuencia son auxiliadas por los servicios médicos forenses del país. En la Figura 2.2 podemos ver como los suicidios

son una problemática que vivimos en nuestro país desde los años 70 que se tiene registro donde vemos cómo de 1970 a 1979 se tiene registros por debajo de las 200 muertes en total de todo el territorio mexicano [51], [52], [53] incrementando los suicidios por arriba de las 650 suicidios (año 1980), donde 558 fueron de hombres y 114 de mujeres, tan solo en ese año, analizando por décadas en 1990, vemos que el total de muertes llegó a 1,405, donde 1,181 muertes fueron de hombres y 224 de mujeres, avanzando otra década en el 2,000 se logra otro incremento de casi el 100%, logrando alcanzar 2,736, donde 2,304 fueron de hombres y 432 de mujeres, lo cual no es extraño que la tendencia en el año 2010 se incrementará llegando a 5,012 muertes y en 2020 otro incremento del casi el 60% lo incrementó el total de muertes hasta llegar a 7,896, afectando nuevamente a los hombres con 6,452 suicidios registrados en comparación con las mujeres con 1,436 suicidios.



Figura 2.2 Agrupación de las Estadísticas de Suicidios en México desde 1970-2021.

Hasta 2005 [51], [52], [53] las estadísticas de intentos de suicidio y suicidios se captaron en las agencias del Ministerio Público; a partir de 2006 se publican datos solamente de suicidios, los cuales se captan mediante los registros de las estadísticas de mortalidad, particularmente, de las accidentales y violentas. Su periodicidad es anual, la información de intentos de suicidio está disponible hasta 2005; la de suicidios de 2006 a 2011; a partir de 2012, los datos se pueden generar mediante las estadísticas de mortalidad. Entre 1980 y 2005 el INEGI en sus reportes declaraba la estadística de intentos de suicidios donde la se visualiza una tendencia inversa a la que se refleja en la Figura anterior, ya que las mujeres con un 25% aproximadamente toman un papel protagónico en comparación con los hombres, en la Figura 2.3 observando en algunos años se tienen similares reportes, pero en los años de 1997 al 2002 se incrementó estos intentos de suicidio por parte de las mujeres.

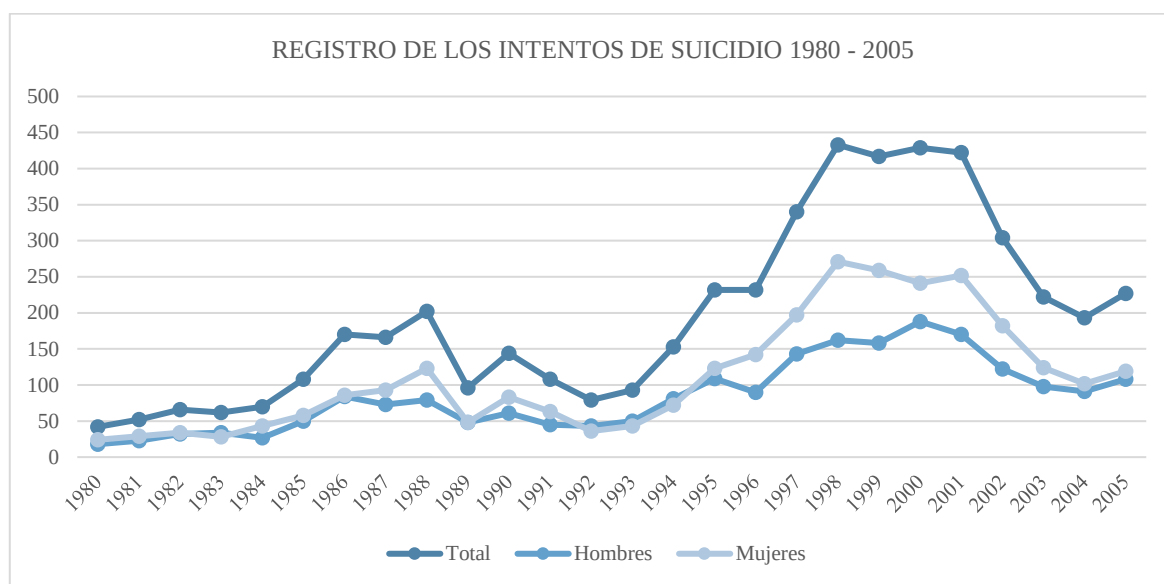


Figura 2.3 Registro de los Intentos de Suicidio 1980 – 2005.

2.5 Suicidios según causa que motivo el acto

En la Tabla 2.1 se detalla la causa que motivo el acto del suicidio, por sexo y año acorde a los datos recolectados de diferentes reportes de INEGI agrupados desde el año 1990 hasta el 2005 [52], donde un dato relevante entre los reportes de estos años es que en 1990 fue el último año que se reportó la “intoxicación por Alcohol y Drogas”.

Tabla 2.1 Ocurrencia y sexo según causa que motivó el acto [52].

CAUSA QUE MOTIVÓ EL ACTO	1990		1995		1996		1998		1999		2001		2002		2003		2005	
	H	M	H	M	H	M	H	M	H	M	H	M	H	M	H	M	H	M
Causa amorosa	69	18	118	38	123	47	140	41	147	48	193	44	186	53	181	65	222	53
Dificultades Económicas	26	5	117	9	81	8	88	6	60	7	76	11	99	13	98	12	102	14
Disgusto familiar	88	40	180	58	179	60	197	61	224	57	209	66	261	70	243	58	241	73
Enfermedad grave e Incurable	85	15	143	28	124	20	111	20	117	26	163	34	146	12	136	34	172	22
Enfermedad mental	50	10	90	32	81	22	75	19	109	25	103	39	87	32	110	30	115	41
Intoxicación por alcohol	62	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Intoxicación por drogas	13	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Remordimiento	17	3	31	9	25	6	32	7	40	8	26	6	25	6	35	7	25	3
Se ignora	71	12	117	23	109	19	123	19	127	17	145	31	152	30	173	30	177	31
	6	3	3	4	3	5	4	4	4	1	8	3	8	3	2	5	8	3
Otra causa	55	8	147	21	147	22	166	23	171	20	306	42	263	56	236	45	315	64

Como se aprecia en la Figura 2.4, una de las causas más relevantes que motivan al suicidio son los “Disgustos familiares”, y se refleja más en las mujeres (70%), seguidas por las “Causas Amorosas” para afecta en ambos sexos. Lo relevante es la clasificación de “Se Ignora”, puesto que la causa que motivó el acto suicida es determinada por los Ministerios Públicos, y al no contar con la información adecuada, estos determinan utilizar dicha clasificación por default en los reportes. A partir del 2006 a la fecha este análisis dejó de ser detallado en los reportes de Estadísticas de Suicidios de los Estados Unidos Mexicanos que son compartidos por el INEGI.

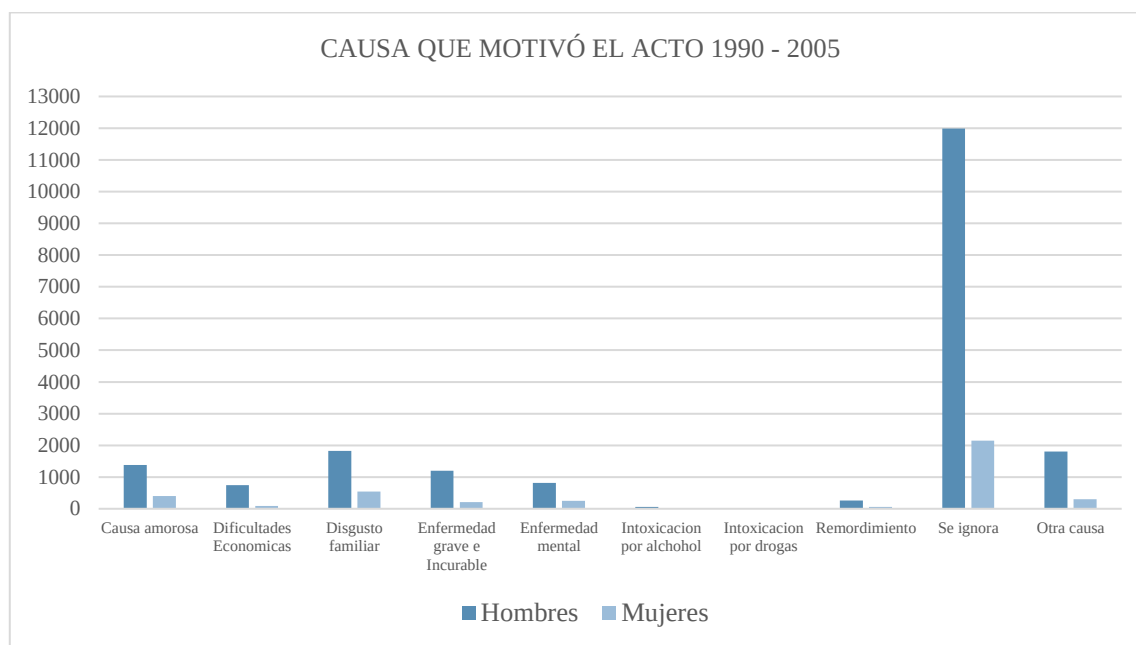


Figura 2.4 Ocurrencia y sexo según causa que motivó el acto [52].

2.6 Suicidio según lugar de ocurrencia en México

Los estudios realizados por el INEGI revelaron que la mayoría de las personas que han cometido el acto suicida ha sido dentro de sus hogares. Además de la facilidad para consumir el acto, también representa un espacio con una carga emocional y simbólica ideal en el proceso de autodestrucción, generando un impacto mayor en los miembros del entorno mediano o familiar. En la Tabla 2.2 podemos observar que la casa habitación es seleccionado como el principal lugar para cometer el suicidio, seguido de la vía pública, así como los suicidios que realizan el campo, hasta el año del 2004 [54] fueron las categorías utilizadas por contabilizar los suicidios, reduciendo el número de categorías a

solo dejar 6 a partir del 2006 y hasta el 2011 [55] que es donde tiene registrado en la estadística de suicidios de los Estados Unidos Mexicanos.

Tabla 2.2 Suicidio según lugar de ocurrencia en México [54].

Lugar del Suicidio	1990-1994	1995	1996	1998	1999	2001	2002	2003	2005
Casa habitación	1037	1837	1735	1843	1987	2339	2465	2590	2780
Hotel o casa de Huéspedes	18	58	30	42	22	48	39	42	37
Vía Publica	96	132	131	167	174	194	195	226	191
Edificio Publico	13	28	36	27	20	33	18	18	28
Cantina, cabaret, restaurant o café	5	9	8	4	3	11	15	4	7
Hospital, Sanatorio o Consultorio	12	28	15	16	18	23	16	23	29
Fabrica o Taller	12	21	19	23	21	29	27	20	27
Campo	113	146	147	140	131	147	158	152	173
Cárcel	29	41	41	43	48	56	54	67	66
otro Lugar	70	128	71	109	107	140	135	147	146
Se Ignora	0	0	0	0	0	67	38	38	69

En la Figura 2.5, se puede observar que el lugar utilizado para el suicidio sigue siendo la vivienda particular o casa habitación como se visualizó en la tabla anterior, seguido de Calles o carreteras que podría considerarse como vía pública.

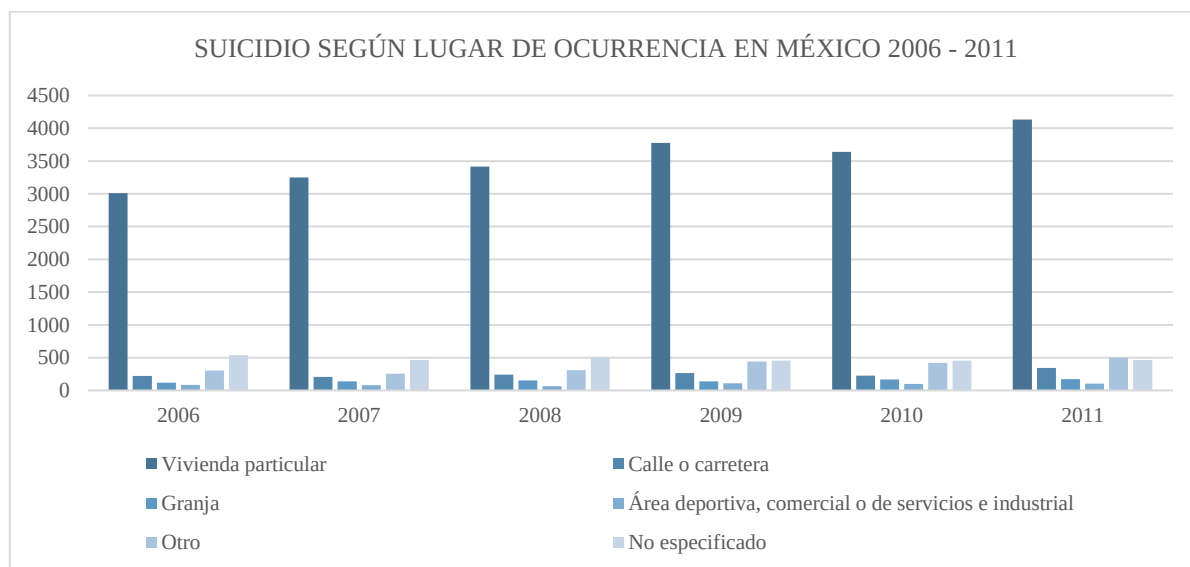


Figura 2.5 Suicidio según lugar de ocurrencia en México 2006 – 2011 [55].

2.7 Suicidio según el medio empleado en México

El medio empleado para el suicidio ha tenido algunas adaptaciones en los nombre de las causas, integrando y eliminado algunas clasificaciones en el transcurso del tiempo, en la Tabla 2.3, se muestran los datos desde 2010 al 2021, donde se observa que los suicidios registrados están clasificados con base en la Clasificación Internacional de Enfermedades (CIE-10), Lesiones autoinfligidas intencionalmente (X60-X84), donde el Ahorcamiento es el medio más utilizado, seguido por los disparos de arma de fuego, que no tiene el mismo crecimiento exponencial como la clasificación anterior, se puede observar un promedio de 532 suicidios entre el año 2010 al 2021 [56].

Tabla 2.3 Medio empleado para el suicidio 2010-2021 [56].

Medio Empleado	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021
Ahorcamiento, estrangulamiento o sofocación	3,847	4,406	4,291	4,554	5,000	5,092	5,145	5,266	5,486	5,992	6,666	7,189
Disparo de arma de fuego	547	595	524	574	529	573	491	518	515	481	532	513
Envenenamiento	460	523	508	556	577	512	517	517	506	477	399	440
Otras causas	158	194	226	225	231	248	217	258	301	273	299	290
Total Suicidios	5,012	5,718	5,549	5,909	6,337	6,425	6,370	6,559	6,808	7,223	7,896	8,432

Los medios más utilizados por hombres y mujeres es el Ahorcamiento, estrangulamiento o sofocación, como puede observar en la Figura 2.6, donde las mujeres prefieren como segunda opción utilizar el envenenamiento como medio a utilizar, a diferencia de los hombres que el segundo método más utilizado por ellos es el Disparo de arma de fuego.



Figura 2.6 Medio utilizado por hombres y mujeres 2010-2021 [56].

La clasificación registrada por el INEGI se determina de la siguiente forma:

- Envenenamiento:
 - analgésicos no narcóticos, antipiréticos y antirreumáticos; drogas antiepilépticas, sedantes, hipnóticas, anti parkinsonianas y psicotrópicas; narcóticos y psicodislépticos (alucinógenos); drogas que actúan sobre el sistema nervioso autónomo; drogas, medicamentos y sustancias biológicas; alcohol; disolventes orgánicos e hidrocarburos halogenados y sus vapores; plaguicidas; productos químicos y sustancias nocivas; y los no especificados.
- Disparo de arma de fuego:
 - Arma corta; rifle, escopeta y arma larga; otras armas de fuego, y las no especificadas.
- Otras causas:
 - Ahogamiento y sumersión; material explosivo; humo, fuego y llamas; vapor de agua, vapores y objetos calientes; objeto cortante; objeto romo o sin filo; saltar desde un lugar elevado; arrojar o colocarse delante de objeto en movimiento; colisión de vehículo de motor; otros medios especificados y no especificadas.

2.8 Suicidios según entidad federativa

De 2010 a 2021, el volumen de suicidios varía entre zonas geográficas y entidades federativas mexicanas [57], en la Figura 2.7 se observar que la entidad federativa que tiene

una constante en estos años son México, Jalisco, Ciudad de México y Guanajuato, en contraparte entidades como Baja California Sur, Nayarit, Colima y Tlaxcala cuentan con menos número de suicidios registrados.

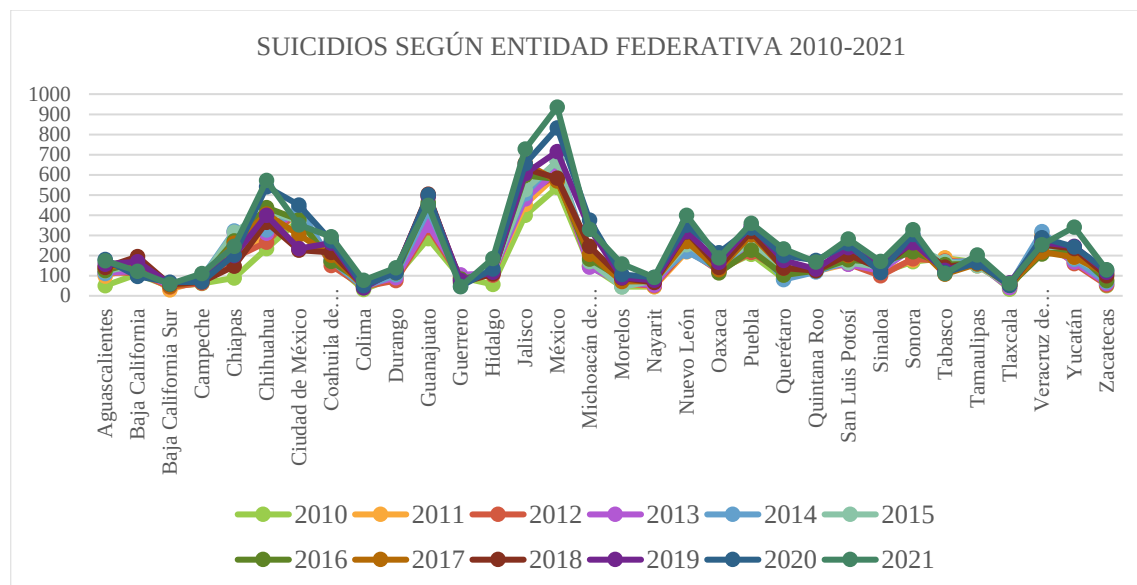


Figura 2.7 Suicidios según entidad federativa 2010-2021 [57].

2.9 Ideación suicida y autolesiones

La estrategia de OMS para la prevención del suicidio parte de la premisa de que este puede ser prevenible, ya que se da como un proceso que inicia con una ideación, donde se puede encontrar características como la preocupación autodestructiva, planificación de un acto letal o un deseo de muerte, que son regularmente posteriores a intentos de suicidio y finalmente a la consumación final del acto. Según la Encuesta Nacional de Salud y Nutrición (ENSANUT) en el 2018 [58], de la población de 10 y más, 5% de esta declaró que alguna vez ha pensado en suicidarse, siendo el 4% detectado entre los hombres y un 6% entre las mujeres. La idea suicida se presenta entre las mujeres mayormente entre edades de 50 a 59 años lo que equivale a un 8%, seguida por las niñas y adolescentes de 10 a 19 años, ver Figura 2.8. En los hombres se presenta mayormente en la población de 40 a 49 años equivalente a un 6%, seguidos del grupo de 30 a 39 y de 60 y más años con una tasa de 5% en cada uno de estos últimos grupos.

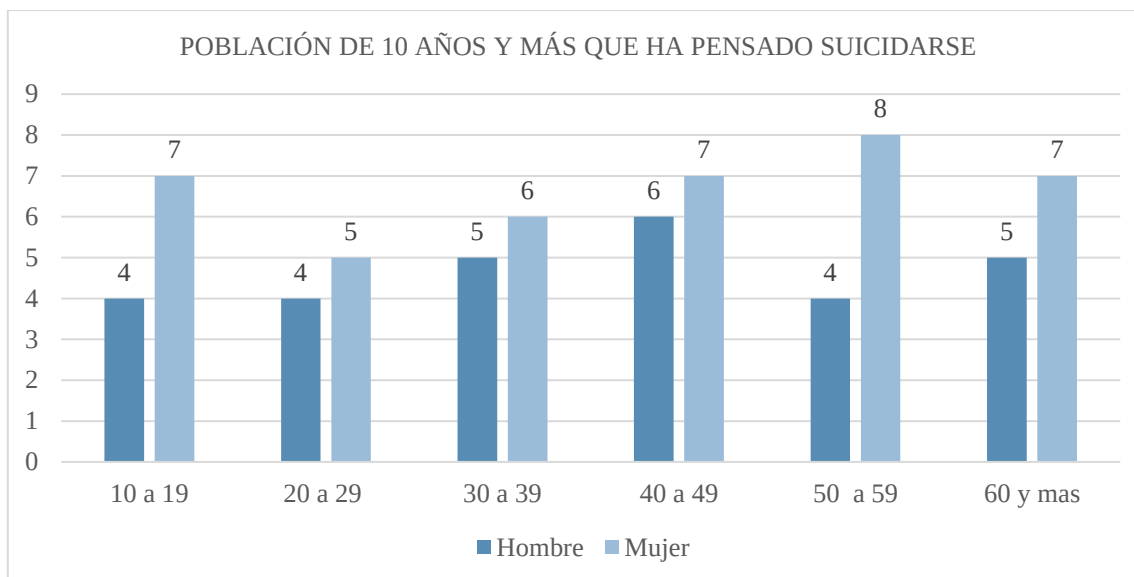


Figura 2.8 Población de 10 años y más que ha pensado suicidarse [58].

La frecuencia de ideación suicida afecta a la población en el grupo de 10 años o más que han pensado suicidarse, el 6% lo hace frecuentemente, el 5% casi siempre, el 34% algunas veces y el 54% rara vez. Esta frecuencia muestra un comportamiento muy similar entre sexos, ya que el 59% de hombres han tenido estos pensamientos suicidas en comparación del 51% de las mujeres, que han pensado suicidarse rara vez, en contraste del 35% de las mujeres y 34% de los hombres lo han pensado algunas veces, ver Figura 2.9.

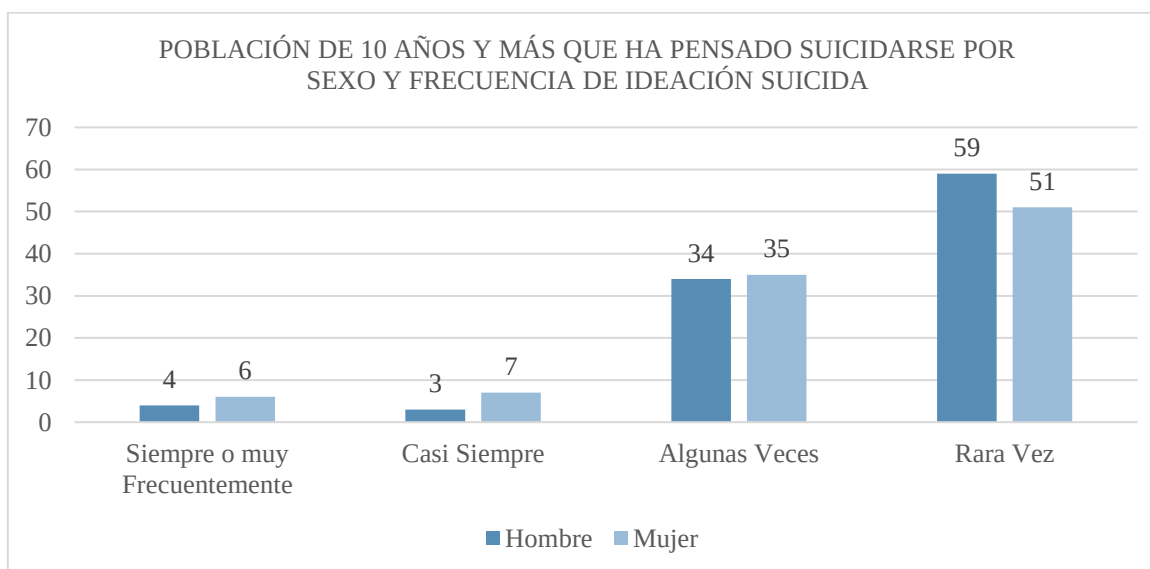


Figura 2.9 Población de 10 años y más que ha pensado suicidarse por sexo y frecuencia de ideación suicida [58].

En la población de 10 y más años, en el estudio declararon que alguna vez a propósito se han herido, cortado, intoxicado o hecho daño con el fin de quitarse la vida, inclinándose mayormente hacia las mujeres con un 3%, en comparación con el 1% de los hombres. Observando que por sexo y grupo de edades destaca el 6% de las mujeres de 10 a 19 años y un 4% entre 20 a 29 años han actuado en autolesiones con el único objetivo de quitarse la vida, en comparativa con el hombre que se presenta en el mismo grupo de edades pero con un 2% de autolesiones [2].

2.10 Factores de riesgo asociados al suicidio

El suicidio es un problema complejo en el que intervienen interdisciplinas que correlacionan múltiples factores tanto psicológicos, ambientales, sociales y biológicos. Pero hay algunas causas muy bien determinadas que influyen en las conductas suicidas. En los niños y adolescentes, influyen especialmente factores como la historia psiquiátrica familiar, enfermedades mentales, la pérdida de un ser querido, la depresión, aislamiento social, abuso de drogas y alcohol. En las mujeres y hombres, suponen un factor muy importante las relaciones con otras personas, la violencia doméstica o el estrés en el ámbito de la familia, sin olvidar enfermedades mentales, abuso de alcohol y drogas, entornos familiares problemáticos. Para el grupo de las personas mayores cuentan especialmente factores como la depresión, el dolor físico a causa de una enfermedad, el aislamiento social y familiar.

2.11 Como prevenir el Acto Suicida

En 2003 la OMS declara al suicidio un problema de salud pública y por tanto prevenible. Se pusieron en marcha los mecanismos para su prevención, estableciendo el día 10 de septiembre el día mundial para la prevención del suicidio. En 2021 la OMS lanzo LIVE LIFE–VIVIR LA VIDA, una guía de implementación que consta de cuatro intervenciones clave para prevenir el suicidio en todo el mundo. La primera intervención es la de velar por el acceso a los medios de suicidio, como puede ser una intervención universal basada en la evidencia para la prevención del suicidio, ya que según el país, se puede prohibir los plaguicidas altamente peligrosos el cual equivale aproximadamente la quita parte de todos los suicidios que se producen en todo el mundo, así como restringir las armas de fuego. La segunda intervención es con los medios de comunicación, para que informen de una forma responsable sobre los suicidios, ya que el número de suicidios por imitación se va incrementando, especialmente si la noticia se refiere a personas famosas. La tercera

intervención es desarrollar aptitudes socioemocionales en los adolescentes, con el objetivo de enfocar con ayuda de otras directrices como la Helping Adolescent Thrive (HAT por sus siglas en inglés) a emplear programas con enfoques positivos de la salud mental, velando por un ambiente escolar seguro. Y por último actuar en una forma oportuna para detectar a tiempo, evaluar, gestionar y hacer seguimiento de personas con algún comportamiento suicida, buscando que esta reciba el apoyo necesario, redireccionando a los trabajadores de la salud y a otras personas, incluidos los familiares para que puedan entrar en contacto inmediato con personas en riesgo [59].

La prevención del suicidio requiere también la intervención de sectores distintos de la salud y exige un enfoque innovador, integral y multisectorial, con la participación tanto del sector de la salud como de otros sectores, como por ejemplo los de la educación, el mundo laboral, la policía, la justicia, la religión, el derecho, la política y los medios de comunicación. El hecho de que los factores de riesgo no inciden en todas las personas por igual, hace necesaria la presencia de los factores protectores, los cuales pueden modular o prevenir el riesgo de la conducta suicida. Entre ellos destaca la configuración de un sentido de la vida y de propósitos elevados.

2.12 Conclusiones del capítulo

En este capítulo se analizó el suicidio como un problema de salud y su prevención es responsabilidad de todos, debido a la gravedad de este problema no sólo en nuestro país sino en todo el mundo. La tasa de mortalidad por suicidio en México ya sea por las características sociodemográfico de quienes intentan suicidarse, el lugar donde cometen el acto, los medios o herramientas que utilizan o el género, esto un patrón alarmante y no debe de tomarse a la ligera. Los datos e información sobre el tema hasta antes de 1970 no existían o no son accesibles e inclusive escaso, estos datos en ocasiones pueden ser no tan acertados, por las irregularidades en los registros de defunción hacen que el suicidio sea problema enigmático. Al ser el suicidio un gran problema social que va en crecimiento exponencial cada año, su correcto tratamiento requiere de un grupo multidisciplinario que busque identificar factores de riesgo para generar modelos de intervención oportuna de prevención que ayude realmente a las personas que estén sufriendo esta problemática. Las personas puede ser clave para la prevención del suicidio, sensibilizándose del problema que está viviendo un amigo, vecino, familiar, etc., ayudándolo a salir adelante, simplemente escuchándolo, sin minimizar las emociones de la persona y sin prejuicios, recomendando se integre a grupos deportivos, sociales y culturales.

Capítulo 3. MACHINE LEARNING

En este capítulo exploramos aspectos de la teoría del aprendizaje automático, también conocido como Machine Learning (ML), que es un subconjunto de la inteligencia artificial (IA). ML es una rama de estudio en la que los algoritmos no necesitan programarse explícitamente, sino que permiten directamente entrenar a las computadoras con conjuntos de datos específicos y el uso de análisis estadístico para predecir los resultados. A partir de ellos se construyen modelos que a su vez permiten que los sistemas informáticos participen en el proceso de toma de decisiones. El concepto de ML se basa en la idea de que un programa de computadora puede acceder a él, entrenarlo y usarlo para aprender para procesos futuros. ML es una de las tecnologías más utilizadas y se puede utilizar en cualquier dominio, como el reconocimiento óptico de caracteres que ayuda a convertir texto en tipos móviles, predicción del comportamiento del usuario, reconocimiento facial y sistemas de recomendación para identificar los intereses de los usuarios en varias redes sociales. El ML está dividido en aprendizaje supervisado, no supervisado, por refuerzo, profundo y no debemos dejar de reconocer que aunque la historia solamente ha llegado hasta estos aprendizajes, el aprendizaje automático depara un futuro a las ventajas cuánticas, por lo cual se debe considerar entre estos aprendizajes al joven campo teórico con el aprendizaje automático cuántico (QML). En este capítulo se detallarán algunos de estos aprendizajes y los algoritmos que en ellos se están utilizando.

3.1 Antecedentes de Machine Learning

La historia del aprendizaje automático es rica y se ha desarrollado a lo largo de varias décadas que se detallan a continuación. El aprendizaje automático, aunque se ha popularizado recientemente, existe desde hace décadas. Es difícil precisar la fecha o la persona que lo inventó, ya que fue una combinación del trabajo de diferentes personas que contribuyeron a la disciplina, con diferentes inventos o algoritmos. La historia del aprendizaje automático es rica y se ha desarrollado a lo largo de varias décadas que se detallan a continuación. En 1943, Walter Pitts y Warren McCulloch presentaron el primer modelo matemático de redes neuronales en su trabajo conjunto titulado “A logical calculus of the ideas immanent in nervous activity” [60]. En 1949, Donald Hebb publicó su libro “The organization of behavior” [61], en el que introdujo nuevas teorías sobre la relación entre el comportamiento, las redes neuronales y la actividad cerebral. Esto ha tenido un impacto significativo en el desarrollo del aprendizaje automático, en cómo las máquinas pueden llegar a “aprender” y adaptarse.

En la década del 1950-1960, Arthur Samuel, pionero de la inteligencia artificial, fue el primero en utilizar el término “Machine Learning”, trabajador de IBM, desarrolló un programa capaz de jugar a las damas con los mejores jugadores de su tiempo, implementó mecanismos que permitieron que su algoritmo aprendiera y mejorará constantemente al recordar las posiciones en las que se encontraba [62]. Marvin Minsky y Dean Edmonds construyeron la primera red neuronal artificial que simulaba una rata que buscaba su camino a través de un laberinto. Diseñaron la primera neurocomputadora (40 neuronas), SNARC (Stochastic Neural Analog Reinforcement Computer), con sinapsis que ajustaban sus pesos (medidas de permeabilidades sinápticas) de acuerdo con el éxito en la realización de una tarea específica. La máquina estaba construida con tubos, motores y embragues, y modeló con éxito el comportamiento de una rata en un laberinto en busca de comida. Frank Rosenblatt en 1957 [63], presentó un algoritmo de aprendizaje supervisado que nombró el perceptrón, el cual sirve para clasificación binaria basado en neuronas artificiales. El cual sentó las bases para futuros desarrollos de algoritmos genéticos de aprendizaje automático.

En la década del 1960-1970, Surge el aprendizaje profundo (DL) con la aportación de los científicos Ucranianos Alexey Ivakhnenko y Valenti Lapa [64], que es un subcampo del aprendizaje automático y la inteligencia artificial, que imita la forma en que los humanos adquieren conocimientos aprendiendo a partir de datos sin procesar. En 1967, Thomas Cover y Peter Hart, profesores de la Universidad de Stanford, publicaron un artículo en el que introdujeron el “The nearest neighbor algorithm” [65], utilizado La regla de decisión del vecino más cercano, que asigna a un punto muestral no clasificado la clasificación del más cercano de un conjunto de puntos previamente clasificados, siendo la época del comienzo del reconocimiento de patrones básicos.

En la década del 1980-1990, Kunihiro Fukushima, propuso un modelo de red neuronal para el reconocimiento de patrones visuales, llamado "neocognitrón" [66]. El neocognitrón es una red jerárquica que consta de muchas capas de células y tiene conexiones variables entre las células de las capas contiguas. Puede adquirir la capacidad de reconocer patrones mediante el aprendizaje y puede entrenarse para reconocer cualquier conjunto de patrones. Fue el precursor de las redes modernas como las redes convolucionales, que se utilizan actualmente para el análisis de imágenes. En 1987, Terrence Sejnowski, creó una red neuronal artificial que aprende pronunciar correctamente un poco más de 20.000 palabras en Inglés. El sistema fue llamado NETtalk, y demostró que incluso una red pequeña puede capturar una fracción significativa de las regularidades en la pronunciación del inglés.

En la década del 1990-2000, Vladimir Vapnik y Corinna Cortes desarrollaron Máquinas de Vectores de Soporte (SVM) [67], inicialmente conocida como red de vectores de soporte, La máquina implementa conceptualmente la siguiente idea: los vectores de entrada se asignan de forma no lineal a un espacio de características de muy alta dimensión. En 1990, Robert Shapir y Yoav Freund [68], presentan el algoritmo Boosting, que es una

técnica de aprendizaje en conjunto que se utiliza para disminuir el sesgo y la variación. Buscando mejorar la predicción de un modelo de inteligencia artificial, afirmando que un modelo de aprendizaje débil genera un aprendizaje fuerte. En este lapso también resurgieron con gran auge las Redes Neuronales Artificiales (ANN), aunque las ANN se remontan a la década de 1950, con el desarrollo de algoritmos de entrenamiento eficientes, como el algoritmo de retropropagación (backpropagation) atribuido a Rumelhart, Hilton, y Williams [69].

En la década del 2000-2010, Leo Breiman en 2001, introduce el algoritmo Random Forest [70], el cual es una combinación de predictores de árboles de modo que cada árbol depende de los valores de un vector aleatorio muestreado de forma independiente y con la misma distribución para todos los árboles del bosque. El error de generalización para los bosques converge hasta un límite a medida que aumenta el número de árboles en el bosque. En esta década el auge del aprendizaje profundo se incrementa, las redes neuronales profundas, especialmente las redes neuronales convolucionales (CNN) y las redes neuronales recurrentes (RNN), revolucionaron el campo del aprendizaje automático y la visión por computadora. Redes Generativas Adversarias (GANs) [71], estas últimas propuestas por Ian Goodfellow en 2014, lo que permitió la generación de datos realistas y la mejora de muchas aplicaciones, como la generación de imágenes y la traducción automática.

En la década del 2010 a la actualidad, el aprendizaje automático continúa evolucionando con avances en áreas como el aprendizaje por refuerzo (Reinforcement Learning), el aprendizaje federado (Federated Learning), el procesamiento de lenguaje natural (PLN) y la inteligencia artificial explicativa. A medida que el campo del aprendizaje automático avanza, se siguen desarrollando y refinando nuevos algoritmos y técnicas para abordar una amplia variedad de problemas en campos como la salud, la robótica, la detección de fraudes, la visión por computadora y más. Ejemplos de estos desarrollos han sido lo que grandes empresas como los equipos de Facebook han aportado con herramientas de reconocimiento facial como DeepFace, o la introducción de los primeros coches autónomos circulando por la ciudad sin conductor. Hay muchos otros inventos interesantes creados mediante el aprendizaje automático. Y estamos sólo en el comienzo de su desarrollo.

Otros conceptos que son de suma importancia y que han ido evolucionando en el transcurso del tiempo y que no se deben dejar pasar en la historia de Machine Learning.

Optimización de Hiperparámetros : Los hiperparámetros son configuraciones ajustables del modelo que deben ajustarse para obtener un mejor rendimiento.

Conjuntos de Entrenamiento y Prueba : Los datos se dividen en conjuntos de entrenamiento y prueba para entrenar y evaluar modelos. El conjunto de entrenamiento se utiliza para aprender, mientras que el conjunto de prueba se utiliza para evaluar el rendimiento del modelo en datos no vistos.

Características (Features) : Las características son las variables o atributos que se utilizan para describir los datos de entrada en un problema de aprendizaje automático. La selección adecuada de características es fundamental para el rendimiento del modelo.

Modelos y Algoritmos : Los modelos son representaciones matemáticas que capturan las relaciones entre las características y las etiquetas de salida. Los algoritmos de aprendizaje automático son métodos específicos para entrenar y ajustar estos modelos, como regresión lineal, árboles de decisión, redes neuronales, entre otros.

Evaluación del Modelo : La evaluación del rendimiento del modelo se realiza mediante métricas como precisión, exactitud, sensibilidad, especificidad, error cuadrático medio (MSE), y otras, según el tipo de problema.

Generalización y Sobreajuste : Un modelo generaliza bien a datos no vistos, evitando el sobreajuste (cuando el modelo se ajusta demasiado a los datos de entrenamiento) y el subajuste (cuando el modelo no se ajusta lo suficiente).

3.2 Aprendizaje Supervisado

Esta técnica entrena el algoritmo a partir de los datos que han sido previamente etiquetados (clasificar) por el autor en una forma manual. Este tipo de etiquetas se le asignan manualmente de entre varias categorías (clases) basado en la experiencia del autor en el conocimiento. Estas categorías se utilizan para representar las entidades que definen un conjunto de variables o datos. Una entidad puede ser representada por un objeto en el mundo real como puede ser “Persona” que puede tener una serie de atributos como el nombre, apellido paterno, apellido materno, fecha de nacimiento, nacionalidad, etc. Un reto con este tipo de aprendizaje es sin duda el etiquetar manualmente una gran cantidad de datos, lo cual requiere de mucho esfuerzo humano y en cantidad de tiempo para realizar dicha tarea. Este tipo de aprendizaje puede ser utilizado para problemas de clasificación o problemas de regresión.

3.2.1 Regresión Lineal

La regresión lineal es un algoritmo de aprendizaje automático supervisado que se utiliza para predecir un valor numérico continuo (la variable dependiente) a partir de una o más variables de entrada (las variables independientes). El objetivo de la regresión lineal es encontrar una relación lineal que mejor se ajuste a los datos de entrada y salida. La regresión lineal es un concepto fundamental en estadísticas y matemáticas, uno de los contribuyentes históricos más influyentes fue Sir Francis Galton, científico británico del siglo XIX que realizó investigaciones pioneras en estadística y regresión. Su trabajo influyó en estadísticos posteriores, como Karl Pearson, quien también hizo contribuciones significativas al campo de la regresión [72]. El Algoritmo de Regresión Lineal funciona

de la siguiente forma. Hipótesis Lineal : En la regresión lineal, se asume que la relación entre las variables de entrada (X) y la variable de salida (Y) puede aproximarse mediante una ecuación lineal, que se representa como:

$$Y = b_0 + b_1X_1 + b_2X_2 + \dots + b_n * X_n \quad (3.1)$$

Donde:

$X_1, X_2, \dots, X_n$ son las variables de entrada, Y es la variable de salida que se desea predecir y $b_0, b_1, b_2, \dots, b_n$, son los coeficientes de la ecuación lineal que se deben estimar durante el proceso de entrenamiento.

Durante el proceso de entrenamiento, el algoritmo busca encontrar los valores óptimos para los coeficientes ($b_0, b_1, b_2, \dots, b_n$) que minimicen la diferencia entre las predicciones y los valores reales en el conjunto de entrenamiento. Se utiliza una métrica de pérdida, como el error cuadrático medio (MSE), para medir cuán bien se ajusta la línea de regresión a los datos de entrenamiento. El proceso de optimización implica ajustar los coeficientes utilizando un algoritmo de optimización, como el descenso del gradiente, que actualiza los coeficientes en pequeños pasos en la dirección que reduce la pérdida. Una vez que se han estimado los coeficientes óptimos, el modelo puede utilizarse para realizar predicciones en nuevos datos de entrada. Simplemente se calcula la ecuación lineal con los coeficientes estimados para obtener la predicción de Y.

Formulación Matemática:

Los coeficientes de regresión se pueden estimar utilizando este método:

$$w_1 = \frac{\sum_{i=1}^{|D|} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{|D|} (x_i - \bar{x})^2} \quad (3.2)$$

$$w_0 = \bar{y} - w_1 \bar{x} \quad (3.3)$$

donde $\bar{x}$ es el valor medio de $x_1, x_2, \dots, x_{|D|}$, e $\bar{y}$ es el valor medio de $y_1, y_2, \dots, y_{|D|}$. Los coeficientes w_0 y w_1 , proporcionan buenas aproximaciones a ecuaciones de regresión que de otro modo serían complicadas [73].

3.2.2 Regresión Logística

La Regresión Logística es un algoritmo utilizado principalmente para problemas de clasificación binaria, aunque también puede extenderse a problemas de clasificación multiclase. Autores como Joseph Berkson [74] se le atribuye la introducción de la regresión logística en la década de 1940, así como David Cox que propuso una versión importante conocida como la regresión de Cox, que se utiliza en el análisis de supervivencia y A. R.; Feinstein y George W. Comstock realizaron contribuciones importantes en la aplicación de la regresión logística en la epidemiología. El Algoritmo de Regresión Logística funciona de la siguiente forma.

Comienza con un conjunto de datos de entrada que contiene características (variables independientes) y etiquetas de clase (variable dependiente). Cada instancia de datos se representa como un vector de características, denotado como X y las etiquetas de clase correspondientes como y . Durante la fase de entrenamiento, la Regresión Logística busca aprender una relación entre las características X y las etiquetas de clase y . A diferencia de la regresión lineal, donde se ajusta una línea recta a los datos, en la regresión logística, se ajusta una curva logística (también conocida como sigmoide) a los datos. La función sigmoide es una función logística que transforma la combinación lineal de las características de entrada en un valor entre 0 y 1.

Formulación Matemática:

La ecuación básica para la regresión lineal con múltiples variables independientes es esta definida por la siguiente ecuación [75]:

$$\hat{Y} = \beta_0 + \beta_1 * X_1 + \beta_2 * X_2 + \dots + \beta_n * X_n \quad (3.4)$$

Los componentes de esta ecuación son: a) $\hat{Y}$, es el resultado continuo estimado, b) $\beta_0 + \beta_1 * X_1 + \beta_2 * X_2 + \dots + \beta_n * X_n$ es la ecuación de regresión lineal para las variables independientes que se utilizarán en el modelo.

Donde:

β_0 , es la intersección o el punto en el que la línea de regresión toca el eje Y vertical, considerado como el valor constante.

$\beta_1 * X_1 + \beta_2 * X_2 + \dots + \beta_n * X_n$, es el valor de cada variable independiente (X_i) ponderando por su respectivo coeficiente beta (β). Estos coeficientes beta dan la pendiente de la línea de regresión o cuánto aumenta el resultado por cada aumento de una unidad en el valor de la variable independiente, cuando mayor es el coeficiente beta, más contribuye su correspondiente variable independiente al resultado.

Durante el entrenamiento, el algoritmo de Regresión Logística ajusta los coeficientes β_0 , β_1 , β_2 , ..., β_n de manera que se ajuste lo mejor posible a los datos de entrenamiento, utilizando la función sigmoide:

$$\sigma(\hat{Y}) = 1 / (1 + e^{(-\hat{Y})}) \quad (3.5)$$

Esto se hace utilizando técnicas de optimización como el descenso del gradiente. Una vez que se han entrenado los coeficientes, se pueden usar para hacer predicciones en nuevos datos. Para predecir la clase de una nueva instancia, se calcula el valor de z utilizando los coeficientes aprendidos y se pasa a través de la función sigmoide. El resultado es una probabilidad entre 0 y 1. Si la probabilidad es mayor que un umbral (por lo general 0.5), se clasifica como la clase positiva; de lo contrario, se clasifica como la clase negativa. Se evalúa el rendimiento del modelo utilizando métricas de evaluación como la precisión, la exhaustividad, la F1-score, la curva ROC, etc.

3.2.3 Máquinas de Vectores de Soporte (SVM)

El algoritmo SVM, o Máquinas de Vectores de Soporte en español, es un método de aprendizaje automático supervisado utilizado tanto para tareas de clasificación como de regresión. El autor del algoritmo es Vladimir Vapnik [67], con contribuciones clave que se remontan a la década de 1960. El objetivo principal del SVM en problemas de clasificación es encontrar un hiperplano que mejor separe dos clases de datos en un espacio multidimensional, maximizando la distancia entre el hiperplano y los puntos de datos más cercanos a él, llamados vectores de soporte. El vector de soporte se refiere a los puntos de datos que se encuentran más cerca de este hiperplano y son fundamentales para la construcción del modelo SVM. Estos vectores de soporte son los que ayudan a definir el margen entre las clases y, por lo tanto, son cruciales para la eficacia del clasificador. El proceso de entrenamiento de una SVM implica encontrar el hiperplano óptimo que maximiza el margen entre las clases mientras minimiza la clasificación errónea de los puntos de datos. Las SVM pueden ser utilizadas con diferentes tipos de kernels (como lineales, sigmoidea, polinómicos y Gaussianos o función de base radial (RBF)) para manejar datos linealmente no separables en el espacio original, proyectándolos en un espacio de mayor dimensionalidad donde la separación sea posible. A continuación, se describe cómo funciona este algoritmo. El SVM también implica la optimización de parámetros, como el parámetro de regularización y los parámetros del kernel. Estos parámetros se ajustan durante el entrenamiento para obtener el mejor rendimiento del modelo. Puede haber múltiples líneas/límites de decisión para separar las clases en un espacio de n dimensiones, pero necesitamos encontrar el mejor límite de decisión que ayude a clasificar los puntos de datos. Las dimensiones del hiperplano dependen de los objetos presentes en el conjunto de datos, lo que significa que si hay 2 objetos, entonces el hiperplano será una línea recta, ver Figura 3.1.

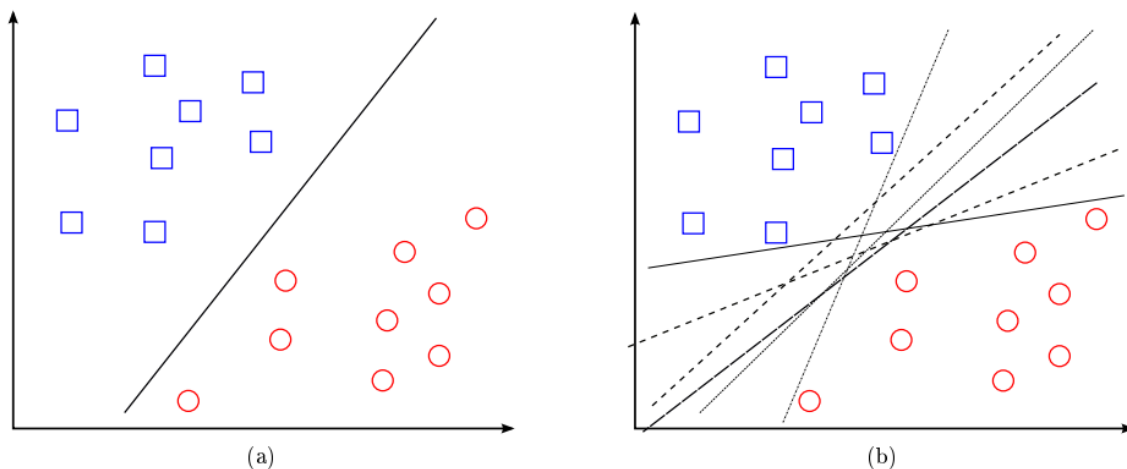


Figura 3.1 Hiperplanos de separación en un espacio bidimensional de un conjunto de ejemplos separables en dos clases [76].

Y si hay 3 signos, entonces el hiperplano será un plano bidimensional. Siempre creamos un hiperplano [76] con margen máximo, lo que significa distancia máxima entre puntos. El margen es la distancia entre el hiperplano de separación y los vectores de soporte más cercanos. SVM busca maximizar este margen, porque un margen amplio sugiere una mejor generalización del modelo a datos no vistos.

Margen estricto (del inglés Hard Margin): El hiperplano de margen máximo o el hiperplano de margen estricto es un hiperplano que separa adecuadamente los puntos de datos de diferentes categorías sin ninguna clasificación errónea.

Margen suave (del inglés Soft Margin): cuando los datos no son perfectamente separables o contienen valores atípicos, SVM permite una técnica de margen suave. Cada punto de datos tiene una variable de holgura introducida por la formulación SVM de margen suave, que suaviza el requisito de margen estricto y permite ciertas clasificaciones erróneas o violaciones.

El núcleo (del inglés Kernel) es la función matemática que se utiliza en SVM para mapear los puntos de datos de entrada originales en espacios de características de alta dimensión, de modo que el hiperplano se pueda encontrar fácilmente incluso si los puntos de datos no son separables linealmente en el espacio de entrada original, la SVM tiene diferentes tipos de kernel, como se observa en Tabla 3.1 con su respectiva función.

Tabla 3.1 Funciones Kernel del SVM

Tipo de Kernel	Función
Lineal	$k_L(x, x') = \langle x, x' \rangle$
Polinómico de grado-p	$k_P(x, x') = [y < x, x' > + \tau]^p, \tau > 0$
Gaussiano	$k_G(x, x') = \exp(-y\ x - x'\ ^2)$ $= \exp(-y\langle (x, x'), (x - x') \rangle), y > 0$
Sigmoidal	$k_S(x, x') = \tanh(y < x, x' > + \tau)$

C (del inglés Cost of misclassification): Las multas por maximización de márgenes y clasificación errónea se equilibran con el parámetro de regularización C en SVM. La sanción por sobrepasar el margen o clasificar erróneamente elementos de datos la decide él mismo. Se impone una penalización más estricta con un valor mayor de C, lo que resulta en un margen menor y quizás menos clasificaciones erróneas.

$$f(w, \varepsilon) = \frac{1}{2} \|w\|^2 + c \sum_{i=1}^n \varepsilon_i \quad (3.6)$$

Pérdida de bisagra (del inglés Hinge Loss): La pérdida de bisagra sanciona clasificaciones incorrectas o violaciones de márgenes. La función objetivo en SVM frecuentemente se forma combinándola con el término de regularización.

Problema dual (del inglés Dual Problem): El problema de optimización que requiere ubicar los multiplicadores de Lagrange relacionados con los vectores de soporte para resolver SVM. La formulación dual permite el uso de trucos del kernel y una computación más efectiva.

Se listan algunas de las ventajas y desventajas del algoritmo SVM que deben ser consideradas a la hora de seleccionar este algoritmo para algún problema en específico.

Ventajas:

- El SVM es eficaz en espacios de características de alta dimensión.
- Puede manejar datos no linealmente separables utilizando kernels.
- Tiene la capacidad de manejar conjuntos de datos desequilibrados.
- SVM funciona mejor cuando los datos son lineales
- Es más efectivo en dimensiones altas.
- Con la ayuda del truco del kernel, podemos resolver cualquier problema complejo.
- SVM no es sensible a valores atípicos
- Puede ayudarnos con la clasificación de imágenes.

Desventajas:

- La elección adecuada del kernel y los parámetros puede ser complicada.
- Puede ser computacionalmente costoso en conjuntos de datos grandes.

Formulación Matemática:

Dados los vectores de entrenamiento $x_i \in \mathbb{R}^p$, en dos clases, y un vector $y \in \{1, -1\}^n$, nuestro objetivo es encontrar $b \in \mathbb{R}$ tal que la predicción dada por $\text{textsign}(w^T \phi(x) + b)$ sea correcta para la mayoría de las muestras. SVC resuelve el siguiente problema primal:

$$\min_{w, b, \zeta} \frac{1}{2} w^T w + C \sum_{i=1}^n \zeta_i$$

$$\text{Sujeto para } y_i(w^T \phi(x_i) + b) \geq 1 - \zeta_i, \quad (3.7)$$

$$\zeta_i \geq 0, i = 1, \dots, n$$

Intuitivamente, estamos tratando de maximizar el margen (*minimizando* $\|w\|^2 = w^T w$), mientras que se incurre en una penalización cuando una muestra está mal clasificada o dentro de la frontera del margen. Idealmente, el valor $y_i(w^T \phi(x_i) + b)$ sería ≥ 1 para todas las muestras, lo que indica una predicción perfecta. Los problemas no suelen ser siempre perfectamente separables con un hiperplano, por lo que permitimos que algunas muestras estén a una distancia ζ_i de su frontera de margen correcta. El término de penalización C controla la fuerza de esta penalización, y como resultado, actúa como un parámetro de regularización inversa. El problema dual al primal es

$$\begin{aligned}
& \min_{\alpha} \frac{1}{2} \alpha^T Q \alpha - e^T \alpha \\
& \text{Sujeeto para} \quad y^T \alpha = 0 \\
& \quad 0 \leq \alpha_i \leq C, i = 1, \dots, n
\end{aligned} \tag{3.8}$$

donde e es el vector de unos, y Q es una matriz semidefinida positiva n por n por $Q_{ij} \equiv y_i y_j K(x_i, x_j)$, donde $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ es el núcleo. Los términos α_i se denominan coeficientes duales y están limitados superiormente por C . Esta representación dual resalta el hecho de que los vectores de entrenamiento son implícitamente mapeados en un espacio dimensional superior por la función ϕ .

Una vez resuelto el problema de optimización, la salida de decisión para una muestra dada x se convierte en: $\sum_{i \in SV} y_i \alpha_i K(x_i, x) + b$, y la clase predicha corresponde a su signo. Sólo se necesita sumar sobre los vectores de soporte porque los coeficientes duales α_i son cero para las demás muestras.

3.2.4 K-Nearest Neighbors (K-NN)

El algoritmo K-Nearest Neighbors (K-NN o k-vecinos más cercanos) [65] es un método de aprendizaje supervisado utilizado principalmente para problemas de clasificación, aunque también se puede aplicar a problemas de regresión. El proceso principal del algoritmo K-NN implica calcular las distancias entre el punto de prueba y todos los puntos del conjunto de entrenamiento, seleccionar los k vecinos más cercanos, contar las etiquetas de clase de estos vecinos y asignar la etiqueta de clase más común al punto de prueba. Este proceso se repite para todos los puntos de prueba, y al final, se devuelven las etiquetas de clase predichas para cada punto de prueba. A continuación, se describe cómo funciona este algoritmo. El algoritmo K-NN, debes elegir un valor para K , que representa el número de vecinos más cercanos que se tendrán en cuenta para tomar una decisión de clasificación o regresión. El algoritmo calcula la distancia entre el punto de datos de prueba (el que deseas clasificar o predecir) y todos los puntos de datos en el conjunto de entrenamiento. Las medidas de distancia comunes incluyen la distancia euclidiana, la distancia de Manhattan y la distancia de Minkowski. Se seleccionan los K puntos de datos del conjunto de entrenamiento que tienen las distancias más cortas con respecto al punto de prueba. Para la clasificación, se toma la clase mayoritaria entre los K vecinos más cercanos como la predicción de la clase del punto de prueba. En el caso de la regresión, se promedian los valores de salida de los K vecinos más cercanos para obtener la predicción de valor continuo.

Formulación Matemática:

El algoritmo KNN supone que todas las instancias corresponden a puntos en el espacio n – dimensional R^n . Los vecinos más cercanos de una instancia a menudo se definen en términos de la distancia euclidiana estándar. Mediante el vector $(x, f(x))$ donde f es la función conceptual que da el valor de clase $f(x)$ para cada instancia x . La instancia x se describe mediante el vector de características $(a_1(x), a_2(x), \dots, a_n(x))$, donde $a_r(x)$ denota el valor del atributo r de la instancia x . La distancia entre dos instancias x_i y x_j se denota como $d(x_i, x_j)$, dnn. Para obtener la distancia euclidiana estándar se define la siguiente ecuación.

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^n (a_r(x_i) - a_r(x_j))^2} \quad (3.9)$$

El aprendizaje del KNN, puede ser obtener valores discretos (para atributos de clases cualitativas) o de valores reales (para atributos de clases cuantitativas).

Un conjunto de n pares $(x_1, \theta_1), \dots, (x_n, \theta_n)$ este dado, donde los x_i 's, toma valores en un espacio métrico X sobre el cual se define una métrica d , y los θ_i 's, toman valores en el conjunto $\{1, 2, \dots, M\}$. Cada θ_i se considera el índice de la categoría a la que pertenece el i -ésimo individuo, y cada x_i es el resultado del conjunto de mediciones realizadas sobre ese individuo. Por brevedad, diremos frecuentemente “ x_i pertenece a θ_i ”, cuando queremos decir precisamente que el i -ésimo individuo en el que se han observado mediciones x_i pertenece a la categoría θ_i . Se da un nuevo par (x, θ) , donde sólo la medición x es observable por el estadístico, y se desea estimar θ utilizando la información contenida en el conjunto de puntos correctamente clasificados. Debemos de llamar: $X'_n \in \{x_1, x_2, \dots, x_n\}$ un vecino más cercano a x_i si $\min d(x_i, x) = d(x'_i, x)$ $i = 1, 2, \dots, n$. La regla del vecino más cercano decide que x pertenece a la categoría θ'_i de su vecino más cercano x'_i . Se comete un error si $\theta'_i \neq \theta$. Observe que la regla NN utiliza sólo la clasificación del vecino más cercano. Las $n - 1$ clasificaciones θ_i restantes se ignoran, ver Figura 3.2 donde se describe el algoritmo de KNN [77].

Algorithm 8.1 Basic k NN Algorithm

Input : D , the set of training objects, the test object, z , which is a vector of attribute values, and L , the set of classes used to label the objects

Output : $c_z \in L$, the class of z

foreach object $y \in D$ **do**

 | Compute $d(z, y)$, the distance between z and y ;

end

Select $N \subseteq D$, the set (neighborhood) of k closest training objects for z ;

$c_z = \operatorname{argmax}_{v \in L} \sum_{y \in N} I(v = \text{class}(y))$;

where $I(\cdot)$ is an indicator function that returns the value 1 if its argument is true and 0 otherwise.

Figura 3.2 Descripción del algoritmo KNN [77].

Se debe de considerar cuestiones claves que afectan el rendimiento del KNN. Una de estas es la elección del k , se puede observar en la Figura 3.3, que muestra un objeto de prueba sin etiquetar X, y objetos de entrenamiento que pertenecen a una clase “*” o “-“. Si k es demasiado pequeño, entonces el resultado puede ser sensible a los de ruido, por otro lado, si k es demasiado grande, entonces la vecindad puede incluir demasiados puntos de otras clases. Se podría obtener una estimación del mejor valor de k mediante una validación cruzada. Sin embargo es importante señalar que $k = 1$ puede ser capaz de realizar otros valores de k , particularmente para conjunto de datos pequeños, sin embargo, en conjuntos de datos más grandes k puede resistir más al ruido.

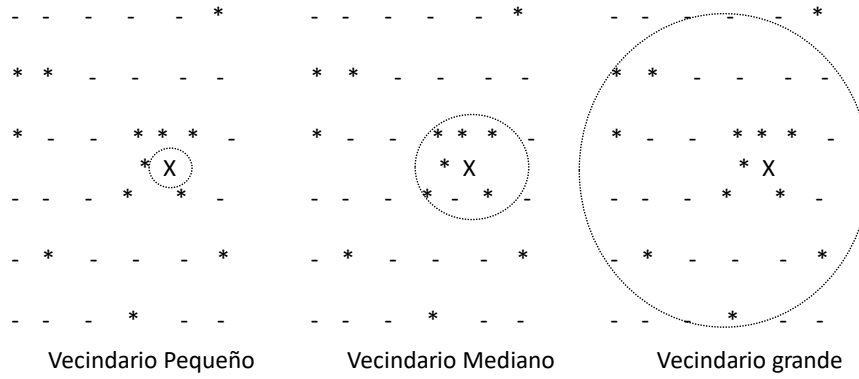


Figura 3.3 KNN clasificador con k pequeño, mediano y grande.

El enfoque para combinar etiquetas de clases es otro de los problemas que se tiene cuando se utiliza este algoritmo. El método más simple para corregir esto, es tomar una votación mayoritaria, pero esto puede ser un problema si los vecinos más cercanos varían mucho en su distancia e indican de manera más confiable la clase del objeto. Un enfoque más acertado para evitar este problema suele resolverse teniendo mucho menos sensibilidad a la elección del k , ponderando el voto de cada objeto por su distancia, esto puede ser resuelto con la siguiente ecuación:

$$c_z = \underset{v \in L}{argmax} \sum_{y \in N} w_i \times I(v = class(c_y)) \quad (3.10)$$

3.2.5 Multi-layer Perceptron (MLP)

El Perceptrón Multicapa (Multi-layer Perceptron, MLP) es una arquitectura de red neuronal artificial (RNA) que se utiliza en el campo del aprendizaje profundo para resolver problemas de clasificación y regresión. A diferencia de algoritmos de aprendizaje supervisado más simples como la regresión logística, MLP es una red neuronal de

múltiples capas que puede aprender representaciones más complejas y no lineales de los datos. El MLP se puede atribuir el resurgimiento y popularización en el campo del aprendizaje profundo a investigadores y científicos como Geoffrey Hinton, Yann LeCun y Yoshua Bengio, quienes han contribuido significativamente al desarrollo de arquitecturas de redes neuronales profundas y algoritmos de aprendizaje profundo. A continuación, se describe cómo funciona este algoritmo.

El MLP es un algoritmo de aprendizaje supervisado que aprende una función $f(\cdot): R^m \rightarrow R^o$ mediante el entrenamiento en un conjunto de datos, donde m es el número de dimensiones de entrada y o es el número de dimensiones de la salida. Dado un conjunto de características $X = x_1, x_2, \dots, x_m$ y un objetivo y , puede aprender un aproximador de función no lineal para la clasificación o la regresión. Se diferencia de la regresión logística en que, entre la capa de entrada y la de salida, puede haber una o más capas no lineales, llamadas capas ocultas. La Figura 3.4 muestra un MLP de una capa oculta con salida escalar.

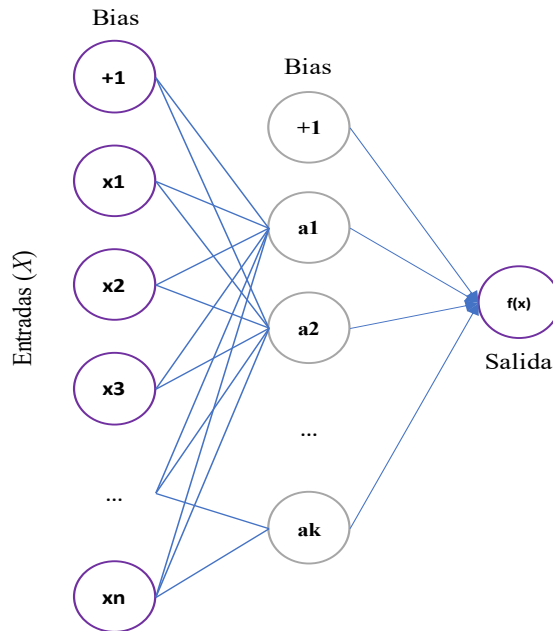


Figura 3.4 Capa Oculta de Algoritmo MLP

La capa más a la izquierda, conocida como la capa de entrada, consiste en un conjunto de neuronas $\{x_i | x_1, x_2, \dots, x_m\}$ que representan las características de entrada. Cada neurona en la capa oculta transforma los valores de la capa anterior con una suma lineal ponderada $w_1x_1 + w_2x_2 + \dots + w_mx_m$, seguida de una función de activación no lineal $g(\cdot): R \rightarrow R$ como la función tangente hiperbólica.

La capa de salida recibe los valores de la última capa oculta y los transforma en valores de salida. Retropropagación del Error (Backpropagation) implementa un algoritmo de

perceptrón multicapa (MLP) que se entrena utilizando los gradientes de la función de pérdida con respecto a los pesos y sesgos de la red. Estos gradientes se utilizan para ajustar los parámetros en la dirección que minimiza la pérdida. El MLP utiliza un algoritmo de aprendizaje supervisado, utilizando el descenso de gradiente estocástico (Stochastic Gradient Descent), Adam, o L-BFGS. El Descenso Gradiente Estocástico (SGD) actualiza los parámetros utilizando el gradiente de la función de pérdida con respecto a un parámetro que necesita adaptación, es decir.

$$w \leftarrow w - \eta \left(\alpha \frac{\partial R(w)}{\partial w} + \frac{\partial Loss}{\partial w} \right) \quad (3.11)$$

Donde η es la tasa de aprendizaje que controla el tamaño del paso en la búsqueda del espacio de parámetros. $Loss$ es la función de pérdida utilizada para la red. Adam es similar al SGD en el sentido de que es un optimizador estocástico, pero puede ajustar automáticamente la cantidad para actualizar los parámetros basándose en las estimaciones adaptativas de los momentos de orden inferior. Con SGD o Adam, el entrenamiento admite aprendizaje en línea y en mini lotes. L-BFGS es un solucionador que aproxima la matriz Hessiana que representa la derivada parcial de segundo orden de una función. Además, aproxima la inversa de la matriz Hessiana para realizar actualizaciones de parámetros [78].

Formulación Matemática:

Dado un conjunto de ejemplos de entrenamiento $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ donde $x_i \in \mathbf{R}^n$ y $y_i \in \{0,1\}$, un MLP de una capa y una neurona ocultas aprende la función:

$$f(x) = W_2 g(W_1^T x + b_1) + b_2 \quad (3.12)$$

donde $W_1 \in \mathbf{R}^m$ son parámetros del modelo. W_1, W_2 representan los pesos de la capa de entrada y de la capa oculta, respectivamente; y b_1, b_2 representan el sesgo añadido a la capa oculta y a la capa de salida, respectivamente. $g(\cdot): \mathbf{R} \rightarrow \mathbf{R}$ es la función de activación, establecida por defecto como la tangente hiperbólica. Está dado como:

$$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}. \quad (3.13)$$

Para la clasificación binaria, $f(x)$ pasa por la función logística:

$$g(z) = 1/(1 + e^{-z}) \quad (3.14)$$

para obtener valores de salida entre cero y uno. Un umbral, fijado en 0,5, asignaría muestras de salida mayores o iguales 0,5 a la clase positiva, y el resto a la clase negativa. Si hay más de dos clases, $f(x)$ sería un vector de tamaño (n_classes,). En lugar de pasar por la función logística, pasa por la función softmax, que se escribe como:

$$softmax(z)_i = \frac{\exp(z_i)}{\sum_{l=1}^k \exp(z_l)} \quad (3.15)$$

donde z_i representa el i -ésimo elemento de la entrada a softmax, que corresponde a la clase i , y K es el número de clases. El resultado es un vector que contiene las probabilidades de que la muestra x pertenece a cada clase. La salida es la clase con la mayor probabilidad. En regresión, la salida permanece como $f(x)$ por lo tanto, la función de activación de salida es solo la función identidad. El MLP utiliza diferentes funciones de pérdida dependiendo del tipo de problema. La función de pérdida para la clasificación es la entropía cruzada (Cross-Entropy), que en el caso binario se da como:

$$Loss(\hat{y}, y, W) = -y \ln \hat{y} - (1 - y) \ln (1 - \hat{y}) + \alpha ||W||_2^2 \quad (3.16)$$

donde $\alpha ||W||_2^2$ es un término de regularización L2 (también conocido como penalización) que penaliza los modelos complejos; y $\alpha > 0$ es un hiperparámetro no negativo que controla la magnitud de la penalización. Para la regresión, el MLP utiliza la función de pérdida del Error Cuadrado; escrita como:

$$Loss(\hat{y}, y, W) = \frac{1}{2} ||\hat{y} - y||_2^2 + \frac{\alpha}{2} ||W||_2^2. \quad (3.17)$$

A partir de los pesos aleatorios iniciales, el perceptrón multicapa (MLP) minimiza la función de pérdida al actualizar repetidamente estos pesos. Después de calcular la pérdida, un paso hacia atrás la propaga desde la capa de salida a las capas anteriores, proporcionando cada parámetro de peso con un valor de actualización destinado a disminuir la pérdida.

En el descenso del gradiente, el gradiente $\nabla Loss_W$ de la pérdida con respecto a los pesos se calcula y se deduce de W . Más formalmente, esto se expresa como:

$$W^{i+1} = W^i - \epsilon \nabla Loss_W^i \quad (3.18)$$

donde i es el paso de iteración, y ϵ es la tasa de aprendizaje con un valor mayor que 0.

El algoritmo se detiene cuando alcanza un número máximo preestablecido de iteraciones; o cuando la mejora de la pérdida está por debajo de un número determinado y pequeño.

Algoritmo del Backpropagation:

- (1) Inicializar todos los pesos y *bias* en la red
- (2) Mientras no se satisfacen las condiciones de parada {
- (3) para cada tupla de entrenamiento X en D {
- (4) // Propaga las entradas hacia adelante:
- (5) por cada unidad de la capa de entrada j {
- (6) $O_j = I_j$; // la salida de una unidad de entrada es su entrada actual
- (7) para cada unidad de capa oculta o de salida j {
- (8) $I_j = \sum_i w_{ij} O_i + \theta_j$; //calcular la entrada j con respecto a la capa anterior, i
- (9) $O_j = \frac{1}{1+e^{-I_j}}$; } //calcula la salida de cada unidad j
- (10) // Retropropaga el error:
- (11) para cada unidad j en la capa de salida

- (12) $Err_j = O_j(1 - O_j)(T_j - O_j)$; //calcula el error
- (13) Para cada unidad j en las capas ocultas, desde la última hasta la primera capa oculta
- (14) $Err_j = O_j(1 - O_j) \sum_k Err_k w_{jk}$; //calcula el error con respecto a la siguiente capa superior, k
- (15) Para cada peso en la red w_{ij} {
- (16) $\Delta w_{ij} = (l)Err_j O_i$; // incremento de peso
- (17) $w_{ij} = w_{ij} + \Delta w_{ij}$; } // actualización de pesos
- (18) Para cada *bias* en la red θ_j {
- (19) $\Delta \theta_j = (l)Err_j$; se incrementa el bias
- (20) $\theta_j = \theta_j + \Delta \theta_j$; // actualización del bias
- (21) }

Donde, las entradas están definidas por D , que es el conjunto de datos que consta de tuplas de entrenamiento y sus valores objetivo-asociados. 1 es la tasa de aprendizaje y la salida una red neuronal de entrada.

El error es propagado hacia atrás a través de la actualización de los pesos y los *bias* para reflejar el error de la predicción de la red. Para una unidad j en la capa de salida, el error Err_j es calculado:

$$Err_j = O_j(1 - O_j)(T_j - O_j); \quad (3.19)$$

Donde O_j es la salida actual de la unidad j , T_j es el valor de target conocido de la tupla de entrenamiento proporcionada. Note que $O_j(1 - O_j)$ es la derivada de la función logística. Para calcular el error de una unidad de capa oculta j , se considera la suma ponderada de los errores de las unidades conectadas a la unidad j en la siguiente capa. El error de una unidad de capa oculta j es:

$$Err_j = O_j(1 - O_j) \sum_k Err_k w_{jk} \quad (3.20)$$

Si, w_{jk} es el peso de la conexión de la unidad j a una unidad k en la siguiente capa superior, y Err_k es el error de la unidad k . Los pesos y *bias* se actualizan para reflejar los errores propagados. Los pesos se actualizan mediante las siguientes ecuaciones,

$$\Delta w_{ij} = (l)Err_j O_i \quad (3.21)$$

donde Δw_{ij} es el cambio en el peso w_{ij} :

$$w_{ij} = w_{ij} + \Delta w_{ij} \quad (3.22)$$

La tasa de aprendizaje es una constante que normalmente tiene un valor entre 0 y 1. La retropropagación aprende usando un método de gradiente descendente para buscar un conjunto de pesos que se ajuste a los datos de entrenamiento para minimizar la distancia

cuadrática media entre la predicción de clase de la red y el valor objetivo conocido de las tuplas. La tasa de aprendizaje ayuda a evitar atascos como mínimo local en el espacio de decisión y alienta encontrar el mínimo global. Si la tasa de aprendizaje es demasiado pequeña, entonces el aprendizaje se producirá a un ritmo muy lento. Si la tasa de aprendizaje es demasiado grande, entonces puede ocurrir oscilación entre soluciones inadecuadas. Una regla general es establecer la tasa de aprendizaje a $1/t$, donde t es el número de iteraciones a través del conjunto de entrenamiento hasta el momento.

Los *bias* se actualizan mediante las siguientes ecuaciones, donde $\Delta\theta_j$:

$$\Delta\theta_j = (l)Err_j \quad (3.23)$$

es el cambio en el *bias* θ_j :

$$\theta_j = \theta_j + \Delta\theta_j \quad (3.24)$$

Alternativamente, los incrementos de peso y *bias* podrían acumularse en variables, de modo que los pesos y sesgos se actualicen después de que se hayan presentado todas las tuplas en el conjunto de entrenamiento. Esta última estrategia se llama actualización de época, donde una iteración a través del conjunto de entrenamiento es una época.

3.2.6 Random Forest

El algoritmo Random Forest (RF) es una técnica de aprendizaje supervisado utilizada principalmente para problemas de clasificación y regresión. Se basa en el concepto de conjuntos (ensemble) de árboles de decisión y se caracteriza por su capacidad para mejorar la precisión y la generalización en comparación con un solo árbol de decisión. Se les atribuye principalmente a los aportes de Leo Breiman [70] por su popularización y desarrollo, aunque sus aportaciones son extensiones de varios conceptos previos en el campo del aprendizaje automático y la estadística. Un bosque aleatorio es un clasificador que consta de una colección de clasificadores estructurados en árbol $\{h(x, \theta_k), k = 1, \dots\}$ donde $\{\theta_k\}$ son vectores aleatorios independientes distribuidos idénticamente y cada árbol emite un voto unitario para la clase más popular en la entrada x . A continuación, se describe cómo funciona este algoritmo.

El algoritmo Random Forest crea un conjunto de árboles de decisión, donde cada árbol se construye de forma independiente utilizando un subconjunto aleatorio de los datos de entrenamiento y un subconjunto aleatorio de las características. Cada árbol se entrena utilizando su conjunto de datos y características específicas. Esto significa que cada árbol es una "opinión" independiente sobre cómo clasificar o predecir los datos. En el caso de la clasificación, cuando se debe tomar una decisión sobre la clase de un nuevo punto de datos, cada árbol emite su voto por la clase. La clase que obtiene la mayoría de los votos se toma como la predicción final. En la regresión, se promedian las predicciones de todos los árboles para obtener la predicción final. El conjunto de árboles en un RF tiende a

reducir la varianza y mejorar la generalización en comparación con un solo árbol de decisión. Esto se debe a que los errores individuales de los árboles se promedian o se votan, lo que reduce la influencia de los datos atípicos y el sobreajuste. El RF también proporciona una medida de la importancia de las características, lo que permite identificar qué características son más relevantes para la tarea de clasificación o regresión. El RF tiene varios hiperparámetros que se pueden ajustar, como el número de árboles en el conjunto (n_estimators), la profundidad máxima de los árboles (max_depth), entre otros, para controlar su rendimiento y capacidad de generalización, ver Figura 3.5.

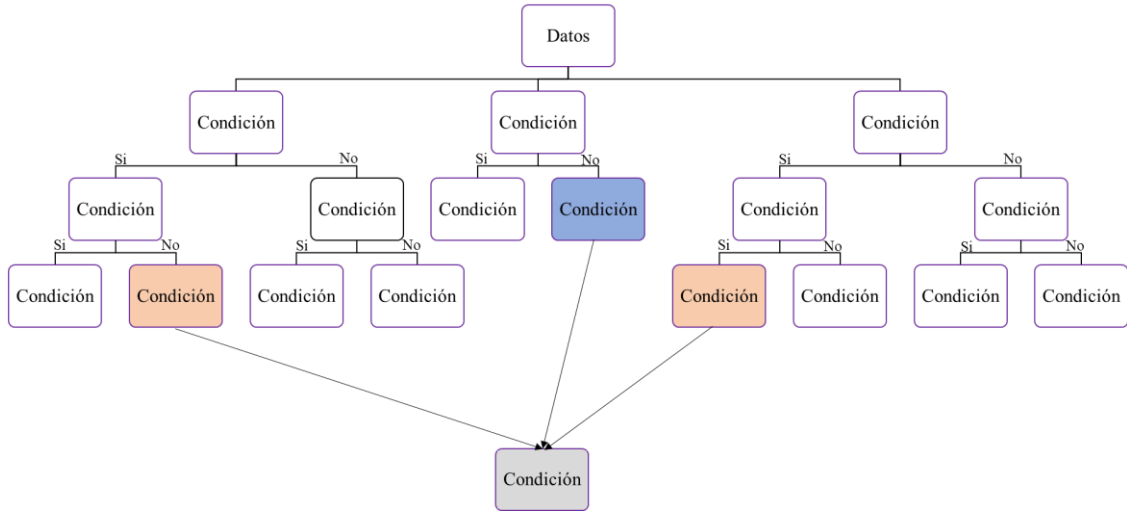


Figura 3.5 Bosques aleatorios (Random Forest)

Formulación Matemática:

Un conjunto de clasificadores $h_1(x), h_2(x), \dots, h_k(x)$, y con el conjunto de entrenamiento extraído al azar de la distribución del vector aleatorio Y, X , defina la función de margen como:

$$ms(X, Y) = av_k l(h_k(X) = Y) - \max_{j \neq Y} av_k l(h_k(X) = j) \quad (3.25)$$

donde $l(\cdot)$ es la función indicadora. El margen mide hasta qué punto el número promedio de votos en X, Y para la clase correcta excede el voto promedio para cualquier otra clase. Cuanto mayor sea el margen, más confianza en la clasificación. El error de generalización viene dado por

$$PE^* = P_{X,Y}(mg(X, Y) < 0) \quad (3.26)$$

donde los subíndices X, Y indican que la probabilidad está sobre el espacio X, Y .

En RF, $h_k(x) = h(X, \theta_k)$. Para una gran cantidad de árboles, de la Ley Fuerte de los Grandes Números y de la estructura del árbol se deduce que:

Teorema 1: A medida que aumenta el número de árboles, es seguro que todas las secuencias $\theta_1 \dots PE^*$ convergerán para:

$$P_{X,Y}(p_\theta(h(X,\theta) = Y) - \max_{j \neq Y} P_\theta(h(X,\theta) = j) < 0) \cdot \quad (3.27)$$

El RF, se puede derivar un límite superior para el error de generalización en términos de dos parámetros que son medidas de cuán precisos son los clasificadores individuales y de la dependencia entre ellos, donde la función de margen del RF se calcula de esta forma:

$$ms(X, Y) = P_\theta(h(X, \theta) = Y) - \max_{j \neq Y} P_\theta(h(X, \theta) = j) < 0 \quad (3.28)$$

Y la fuerza del conjunto de clasificadores $\{h(X, \theta)\}$ es:

$$s = E_{X,Y}mr(X, Y) \quad (3.29)$$

Suponiendo $s \geq 0$, la desigualdad de Chebyshev da:

$$PE^* \leq var(mr) / s^2 \quad (3.30)$$

Una expresión más reveladora para la varianza de mr se obtiene de la siguiente manera:

$$\hat{j}(x, J) = \arg \max_{j \neq Y} P_\theta(h(X, \theta) = j) \quad (3.31)$$

Entonces

$$ms(X, Y) = P_\theta(h(X, \theta) = Y) - P_\theta(h(X, \theta) = \hat{j}(x, J)) = E_\theta[I(h(X, \theta) = Y) - I(h(X, \theta) = \hat{j}(x, J))] \quad (3.32)$$

Para la función de margen bruto (raw en inglés) del RF se define de la siguiente forma:

$$rmgs(\theta, X, Y) = I(h(X, \theta) = Y) - I(h(X, \theta) = \hat{j}(x, J)) \quad (3.33)$$

Por lo tanto, $mr(X, Y)$ es la expectativa de $rmg(\theta, X, Y)$ con respecto a θ . Para cualquier función de la identidad:

$$[E_\theta f(\theta)]^2 = E_{\theta, \theta'} f(\theta) f(\theta') \quad (3.34)$$

Se cumple donde θ, θ' , son independientes con la misma distribución, lo que implica:

$$mr(X, Y)^2 = E_{\theta, \theta'} rmg(\theta, X, Y) rmg(\theta', X, Y) \quad (3.35)$$

Utilizando la función anterior obtenemos:

$$\begin{aligned} var(mr) &= E_{\theta, \theta'} (cov_{X,Y} rmg(\theta, X, Y) rmg(\theta', X, Y)) = \\ &E_{\theta, \theta'} (p(\theta, \theta') sd(\theta), sd(\theta')) \end{aligned} \quad (3.36)$$

Donde $p(\theta, \theta')$ es la correlación entre $rmg(\theta, X, Y)$ y $rmg(\theta', X, Y)$ manteniéndolo θ, θ' fijo y $sd(\theta)$ es la desviación estándar de $rmg(\theta, X, Y)$ manteniendo θ fijo, entonces se obtiene:

$$var(mr) = \bar{p}(E_\theta sd(\theta))^2 \leq \bar{p} E_\theta var(\theta) \quad (3.37)$$

Donde $\bar{p}$ es el valor medio de la correlación; esto es:

$$\bar{p} = E_{\theta, \theta'}(p(\theta, \theta')sd(\theta), sd(\theta'))/E_{\theta, \theta'}(sd(\theta), sd(\theta')) \quad (3.38)$$

Y escribe de esta forma:

$$E_{\theta} var(\theta) \leq E_{\theta}(cov_{X,Y}rmg(\theta, X, Y))^2 \leq 1 - s^2. \quad (3.39)$$

El RF tiene el límite superior para el error de generalización dado por :

$$PE^* \leq \bar{p}(1 - s^2) / s^2 \quad (3.40)$$

El limite en el RF puede ser flexible, este debe de cumplir la misma función sugestiva con los límites del tipo VC para otro tipo de clasificadores. Esto demuestra que los dos ingredientes involucrados en el error del RF son la fuerza de los clasificadores individuales en el bosque y la correlación entre ellos, en términos de la función de margen bruto. La relación c/s^2 , es la correlación dividida por el cuadrado de la fuerza, cuanto más pequeña sea, será mejor el resultado. La relación c/s^2 para el RF se define como:

$$c/s^2 = \bar{p} / s^2 \quad (3.41)$$

En el caso de que la situación del RF tenga dos clases, la función de margen se utilizaría como:

$$mr(X, Y) = 2P_{\theta}(h(X, \theta) = Y) - 1 \quad (3.42)$$

El requisito es que la fuerza sea positiva, se vuelve similar a la conocida condición de aprendizaje débil $E_{X,Y}P_{\theta}(h(X, \theta) = Y) > .5$. La función de margen bruto es $2I(h(X, \theta) = Y) - 1$ y la correlación esta $\bar{p}$ entre $I(h(X, \theta) = Y)$ e $I(h(X, \theta') = Y)$. En particular, si se toma que los valores de Y son +1 y -1, entonces

$$\bar{p} = E_{\theta, \theta'}[p(h(\cdot, \theta), h(\cdot, \theta'))] \quad (3.43)$$

De modo que $\bar{p}$ es la correlación entre los dos miembros de la clase de diferentes bosques que se promedian sobre la distribución. Para más de dos clases la medida de resistencia definida depende del bosque, así como de los árboles individuales, ya que el bosque que determine $\hat{f}(x, j)$ podría tener otro enfoque como:

$$PE^* = P_{X,Y} \left(P_{\theta}(h(X, \theta) = Y) \right) - \max_{j \neq Y} P_{\theta}(h(X, \theta) = j) < 0 \leq \sum_j P_{X,Y} \left(P_{\theta}(h(X, \theta) = Y) \right) - P_{\theta}(h(X, \theta) = j) < 0 \quad (3.44)$$

Que define la fuerza del conjunto de clasificadores $\{h(X, \theta)\}$, en relacion a la clase j.

$$s_j = E_{X,Y} \left(P_{\theta}(h(X, \theta) = Y) - P_{\theta}(h(X, \theta) = j) \right) \quad (3.45)$$

Utilizando la desigualdad Chebyshev, supone que todo $s_{j>0}$ conducen a:

$$PE^* \leq \sum_j var \left(P_{\theta}(h(X, \theta) = Y) - P_{\theta}(h(X, \theta) = j) \right) s_j^2 \quad (3.46)$$

Usando las identidades similares a las utilizadas para derivar las varianzas, se puede expresar en términos de correlación promedio.

3.3 Aprendizaje No Supervisado

En esta técnica el algoritmo utiliza un conjunto de datos que no tienen algún tipo de categorización, en este caso el algoritmo no tiene información de las categorías, lo cual constituye un tipo de aprendizaje por observación y descubrimiento, donde el sistema de aprendizaje analiza las entidades y determina las características que tienen en común entre ellas para poder ser agrupadas y con esto formar una salida conceptual de la información. Los algoritmos que utilizan este tipo de técnicas y que son los más utilizados.

3.3.1 K-Means

El algoritmo K-Medias (K-Means), basado en la división, es un tipo de algoritmo de clúster, y fue propuesto por J.B. MacQueen. Este algoritmo, suele utilizarse en la minería de datos y el reconocimiento de patrones con el objetivo de minimizar el índice de rendimiento de los clústeres, el error cuadrado y el criterio de error son los fundamentos de este algoritmo. Para buscar el resultado óptimo, este algoritmo trata de encontrar K divisiones para satisfacer un determinado criterio. Este elige algunos puntos para representar los puntos focales del clúster inicial, seguido de reunir los puntos muestra restantes de sus puntos focales de acuerdo con el criterio de la distancia mínima [79]. A continuación, se describe cómo funciona este algoritmo.

Inicialización:

El algoritmo k-medias divide un conjunto N de X muestras en K conglomerados disjuntos C , cada uno descrito por la media μ_j de las muestras en el conglomerado. Las medias se denominan comúnmente los «centroides» del conglomerado. El algoritmo K-medias tiene como objetivo elegir los centroides que minimicen la inercia, o el criterio de la suma de cuadrados dentro del conglomerado:

$$\sum_{i=0}^n \min_{\mu_j \in C} (||x_i - \mu_j||^2) \quad (3.47)$$

La inercia puede reconocerse como una medida de la coherencia interna de los conglomerados. La inercia supone que los conglomerados son convexos e isotrópicos, lo que no siempre es el caso. Responde mal a los conglomerados alargados o a las variedades con formas irregulares. La inercia no es una métrica normalizada, es decir los valores más

bajos son mejores y cero es óptimo. Pero en espacios de muy alta dimensión, las distancias euclidianas tienden a inflarse, observar la Figura 3.6.

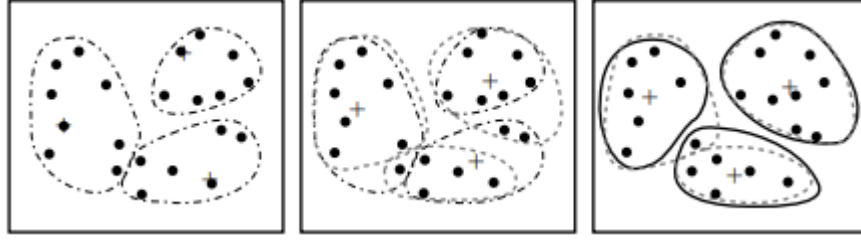


Figura 3.6 Medida de Inercia en Algoritmo K-Medias [73].

Formulación Matemática:

Dadas N muestras del patrón $\{x_1, x_2, \dots, x_n\}$, que están esperados ser clasificadas en grupos de K .

- 1) Selecciona cualquier entre $\{x_1, x_2, \dots, x_n\}$ para que actúe como el primer punto focal del grupo z_1 se selecciona $z_1 = x_1$
- 2) Selecciona otro punto que este lo más alejado posible de z_1 para que sea el otro punto focal para ser el segundo para agrupar y calcular la distancia entre cada ejemplo y z_1 .
 $\|x_{1-z_1}\|, i = 1, 2, \dots, N$, si:

$$\|x_{j-z_1}\| = \max\{\|x_{j-z_1}\|, i = 1, 2, \dots, N\}, j = 1, 2, \dots, N \quad (3.48)$$

Entonces se elige x_j como punto focal del segundo grupo $z_2 = x_j$.

- 3) Calcula la distancia entre cada muestra entre $\{x_1, x_2, \dots, x_n\}$ y $\{z_1, z_2\}$ una por una.

$$d_{i1} = \|x_i - z_1\|, i = 1, 2, \dots, N \quad (3.49)$$

$$d_{i2} = \|x_i - z_2\|, i = 1, 2, \dots, N \quad (3.50)$$

Escogiendo el mínimo de los resultados $\min(d_{i1}, d_{i2}), i = 1, 2, \dots, N$. Reuniendo los mínimos de todas las muestras del patrón $\{z_1, z_2\}$. Se elige el máximo entre los mínimos para que sea el tercer punto focal del grupo z_3 , si

$$\min(d_{j1}, d_{j2}) = \max\{\min(d_{i1}, d_{i2}), i = 1, 2, \dots, N, j = 1, 2, \dots, N\} \quad (3.51)$$

Entonces $z_3 = x_j$.

- 4) Se tiene $r (r < k)$, como puntos focales del grupo $\{z_i, i = 1, 2, \dots, r\}$, ahora se necesita determinar el punto focal del grupo $r+1$ es decir, si: $\min(d_{j1}, d_{j2} \dots d_{jr}) = \max\{\min(d_{j1}, d_{j2} \dots d_{jr}), i = 1, 2, \dots, N\} j = 1, 2, \dots, N$ entonces $z_{r+1} = x_j$.
- 5) Repetir hasta que $r+1 = K$

- 6) Al tener elegido el punto focal K del grupo inicial $z_1(1), z_2(1), \dots, z_k(1)$. Los números entre paréntesis son los numero utilizados en operaciones iterativas para buscar los puntos de agrupación.
- 7) Siguiendo la regla de minimizar la distancia, se asigna $\{x_1, x_2, \dots, x_n\}$ a uno de los Grupos K , es decir si:

$$\|x - z_j(t)\| = \min\{\|x - z_i(t)\|, i = 1, 2, \dots, K\}, j = 1, 2, \dots, K \quad (3.52)$$

- 8) Calculando los nuevos valores vectoriales de cada punto focal del grupo: $z_j(t + 1)$, $j = 1, 2, \dots, K$, con esto se calcula los vectores medios de muestras de cada grupo:

$$z_j(t + 1) = \frac{1}{N_j} \sum_{x \in s_j(t)} x, j = 1, 2, \dots, K \quad (3.53)$$

El símbolo N_j en la fórmula anterior representa el número de muestras del jésimo grupo s_j . Se calculan los vectores medios de muestras de los K grupos respectivamente. Esto hace que los vectores medios sean nuevos grupos puede minimizar la función de criterio de grupo J_j como:

$$J_j = \sum_{x \in s_j(t)} \|x - z_j(t + 1)\|, j = 1, 2, \dots, K \quad (3.54)$$

- 9) Si, $z_j(t + 1) \neq z_j(t), j = 1, 2, \dots, K$ entonces se tiene que retornar al punto 7), se vuelve a clasificar las muestras del patrón una por una y se repiten las operaciones iterativas. Si $z_j(t + 1) = z_j(t), j = 1, 2, \dots, K$, entonces la convergencia del algoritmo ha terminado.

3.3.2 Agrupamiento Jerárquico

El algoritmo de Agrupamiento Jerárquico es un método no supervisado que construye conglomerados anidados fusionándose o dividiéndolos sucesivamente. Esta jerarquía de conglomerados se representa como un árbol o dendrograma [80], que muestra la relación entre los grupos en diferentes niveles de granularidad, ver Figura 3.7.

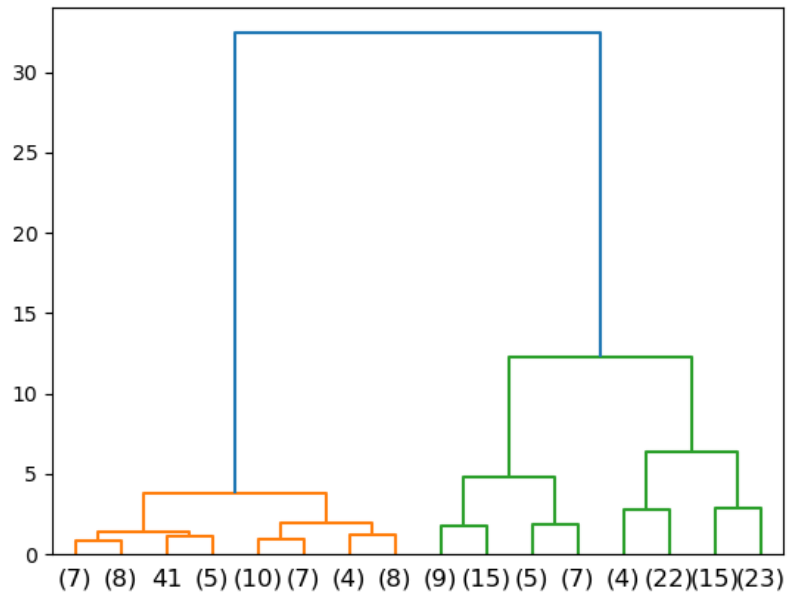


Figura 3.7 Fusión jerárquica de conglomerados [67].

Tipos de Enlazamientos:

Ward: Minimiza la suma de las diferencias al cuadrado dentro de todos los conglomerados.

Máximo o enlazamiento completo: Minimiza la distancia máxima entre las observaciones de pares de conglomerados.

Enlazamiento promedio: Minimiza el promedio de las distancias entre todas las observaciones de pares de conglomerados.

Enlazamiento simple: Minimiza la distancia entre las observaciones más cercanas de los pares de conglomerados.

Inicialización: Cada punto de datos se considera un clúster individual en el nivel más bajo de la jerarquía. Los clústeres se combinan iterativamente en función de su similitud. Los dos clústeres más similares se fusionan en un nuevo clúster, que se convierte en un nodo en el árbol jerárquico. El proceso de fusión se repite hasta que todos los puntos de datos estén agrupados en un solo clúster en la parte superior del árbol jerárquico. El resultado es un dendrograma que muestra la estructura jerárquica de los clústeres, donde se pueden identificar clústeres a diferentes niveles de granularidad. Según los objetivos del análisis, se pueden seleccionar diferentes niveles de la jerarquía para obtener los clústeres deseados.

Formulación Matemática:

Si C es una asignación de la clase basada en la evidencia y K el agrupamiento, definamos a y b como:

- a , el número de pares de elementos que están en el mismo conjunto en C y en el mismo conjunto en K
- b , el número de pares de elementos que están en conjuntos diferentes en C y en conjuntos diferentes en K

El índice de Rand no ajustado viene dado entonces por:

$$RI = \frac{a+b}{C_2^{n_{samples}}} \quad (3.55)$$

donde $C_2^{n_{samples}}$ es el número total de pares posibles en el conjunto de datos. No importa si el cálculo se realiza sobre pares ordenados o pares no ordenados siempre y cuando el cálculo se realice de forma coherente. El índice de Rand no garantiza que las asignaciones aleatorias de etiquetas obtengan un valor cercano a cero. Para contrarrestar este efecto podemos descontar el RI esperado $E[RI]$ de las etiquetas aleatorias definiendo el índice de Rand ajustado de la siguiente manera:

$$ARI = \frac{RI - E[RI]}{\max(RI) - E[RI]} \quad (3.56)$$

3.3.3 PCA (Análisis de Componentes Principales)

El Análisis de Componentes Principales (PCA) es una técnica de reducción de dimensionalidad ampliamente utilizada en el campo del aprendizaje automático y la estadística. Es un método estadístico que permite simplificar la complejidad de espacios muestrales con muchas dimensiones a la vez que conserva su información. El PCA permite por lo tanto “condensar” la información aportada por múltiples variables en solo unas pocas componentes [81]. Autores como Karl Pearson y Harold Hotelling realizaron contribuciones significativas en este campo a principios del siglo XX. Su objetivo principal es reducir la dimensionalidad de un conjunto de datos manteniendo la mayor cantidad posible de su variabilidad original. A continuación, se describe cómo funciona este algoritmo.

El análisis de componentes principales, o PCA (también llamado método de Karhunen-Loeve o K-L), busca k vectores ortogonales n -dimensionales que se pueden utilizar mejor para representar los datos, donde $k \leq n$. De este modo, los datos originales se proyectan en un espacio mucho más pequeño, lo que da como resultado una reducción de dimensionalidad. A diferencia de la selección de subconjuntos de atributos, que reduce el tamaño del conjunto de atributos al retener un subconjunto del conjunto inicial de atributos, PCA “combina” la esencia de los atributos creando un conjunto alternativo de variables más pequeño. Luego, los datos iniciales se pueden proyectar en este conjunto

más pequeño [73]. La PCA a menudo revela relaciones que antes no se sospechaban y, por lo tanto, permite interpretaciones que normalmente no resultarían, ver Figura 3.8. A continuación, se describe cómo funciona este algoritmo.

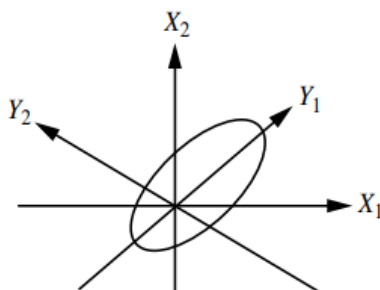


Figura 3.8 Análisis de componentes principales [67].

Los datos de entrada están normalizados, de modo que cada atributo se encuentre dentro del mismo rango. Este paso ayuda a garantizar que los atributos con dominios grandes no dominen a los atributos con dominios más pequeños. El PCA calcula k vectores ortonormales que proporcionan una base para los datos de entrada normalizados. Estos son vectores unitarios que apuntan cada uno en una dirección perpendicular a los demás. Estos vectores se denominan componentes principales. Los datos de entrada son una combinación lineal de los componentes principales. Los componentes principales se clasifican en orden de “importancia” o fuerza decreciente. Los componentes principales sirven esencialmente como un nuevo conjunto de ejes para los datos, proporcionando información importante sobre la varianza. Es decir, los ejes ordenados son tales que el primer eje muestra la mayor varianza entre los datos, el segundo eje muestra la siguiente varianza más alta, y así sucesivamente. Por ejemplo, la muestra de los dos primeros componentes principales, Y_1 e Y_2 , para el conjunto de datos dado originalmente asignado a los ejes X_1 y X_2 . Esta información ayuda a identificar grupos o patrones dentro de los datos. Debido a que los componentes se clasifican según un orden decreciente de “importancia”, el tamaño de los datos se puede reducir eliminando los componentes más débiles, es decir, aquellos con baja varianza. Utilizando los componentes principales más fuertes, debería ser posible reconstruir una buena aproximación de los datos originales. El PCA es computacionalmente económico, se puede aplicar a atributos ordenados y desordenados y puede manejar datos dispersos y sesgados. Los datos multidimensionales de más de dos dimensiones se pueden manejar reduciendo el problema a dos dimensiones. Los componentes principales pueden utilizarse como insumos para la regresión múltiple y el análisis de conglomerados.

Formulación Matemática:

El PCA que utilizan la varianza y la covarianza las cuales se aplican en los siguientes métodos:

$$var(X) = \frac{\sum_{i=1}^n (X_i - \bar{x})(x_i - \bar{x})}{(n-1)} \quad (3.57)$$

$$cov(X, Y) = \frac{\sum_{i=1}^n (X_i - \bar{x})(x_i - \bar{x})}{(n-1)}, \quad (3.58)$$

Métodos basados en Correlación: Es te método parte de la matriz de correlación que es cuando los datos no son dimensionalmente homogéneos o el orden de magnitud de las variables aleatorias medidas no es el mismo. Dado el valor de cada una de la m que son las variables aleatorias F_j , para cada uno de n individuos se tomara el valor de las variables en el conjunto de datos en forma de matriz. $(F_j^\beta)_{j=1, \dots, m}^{\beta=1, \dots, n}$, donde cada conjunto $M_j = \{F_j^\beta \mid \beta = 1, \dots, n\}$. Puede ser una muestra aleatoria para la variable F_j . A partir de los $m \times n$ datos que correspondientes a las m variables aleatorias, la matriz de correlación muestral se define por:

$$R = [r_{ij}] \in M_{m \times m}, \quad (3.59)$$

donde $r_{ij} = \frac{cov(F_i, F_j)}{\sqrt{var(F_i)var(F_j)}}$ Dado que la matriz es de correlaciones es simétrica entonces el resultado sería una diagonal y sus valores propios de λ_i se verificarían con: $\sum_{i=1}^m \lambda_i = m$, debido que las propiedades de m valores propios reciben el nombre de pesos de cada uno de los m componentes principales, los factores principales serán representados por los vectores propios de la matriz R .

Método basado en la covarianza: Es te método parte de la matriz de covarianza que se usa cuando los datos son dimensionalmente homogéneos y presentan valores medios similares. El objetivo es transformar un conjunto dado de datos X de la dimensión $m \times n$ a otro conjunto de datos Y de menor dimensión $n \times l$ con la menor perdida de información útil. Una parte de conjunto n muestra de cada una de las cuales tienen m variables que las describen con solo l variables, donde $l < m$. El número de componentes principales l tiene que ser inferior a la menor de las dimensiones X , logrando esto con: $l \leq \min\{n, m\}$.

Estos datos deben ser analizados centrándolos a media 0 y auto escalándolos con la siguiente formula.

$$X = \sum_{a=1}^l t_a P_a^t + E \quad (3.60)$$

Los vectores t_a conocidos como scores, que contienen la información de las muestras como han sido relacionadas unas con otras con la propiedad de ser ortogonales. Los vectores P_a son conocidos como loadings e informan la relación existente entre las variables, teniendo la cualidad de ser ortonormales. En la matriz E acumula el error de ajuste del modelo de datos al recoger menos componentes principales que variables. La matriz de covarianza es calculada por el PCA en base a la descomposición en los vectores propios de la covarianza, que utiliza la siguiente formula:

$$cov(x) = \frac{x^T x}{n-1}, cov(x)P_a = \lambda_a P_a, \sum_{a=1}^m \lambda_a = 1 \quad (3.61)$$

donde λ_a es el valor propio asociado al vector P_a , que al final por último realiza $t_a = xP_a$, entendiendo que t_a son las proyecciones de X en P_a donde los valores propios de λ_a miden la cantidad de varianza capturada, que equivale a la información que representa cada uno de los componentes principales, donde estos van disminuyendo según su número, es decir el componente principal número uno representa más información que el dos y así sucesivamente.

3.3.4 t-SNE (t-distributed Stochastic Neighbor Embedding)

El t-SNE es una técnica de reducción de dimensionalidad utilizada principalmente para visualizar datos de alta dimensionalidad en un espacio de menor dimensión (generalmente 2D o 3D). Fue propuesto por Laurens van der Maaten y Geoffrey Hinton. Laurens van der Maaten, en particular, desempeñó un papel clave en el desarrollo y la popularización de t-SNE como una técnica efectiva de reducción de dimensionalidad y visualización. A diferencia de PCA (Análisis de Componentes Principales), t-SNE se enfoca en preservar las relaciones locales entre puntos de datos en el espacio original, lo que lo hace especialmente útil para la visualización [82], ver ejemplo de Figura 3.9. A continuación, se describe cómo funciona este algoritmo.

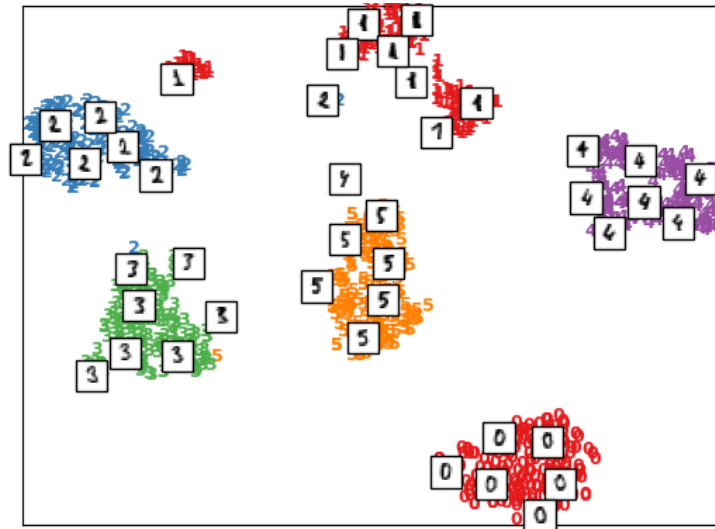


Figura 3.9 t-SNE incrustación de los dígitos [83].

Inicialización: Se construye una distribución de probabilidad que mide las similitudes entre pares de puntos de datos en el espacio de alta dimensión. Esto se hace utilizando una función de similitud, como la función Gaussiana, creando una distribución de probabilidad similar en el espacio de baja dimensión (generalmente 2D o 3D). El t-SNE busca

minimizar la divergencia Kullback-Leibler entre las distribuciones de probabilidad conjuntas en el espacio de alta y baja dimensión. Esto implica ajustar la posición de los puntos en el espacio de baja dimensión de manera que las similitudes entre los puntos se preserven de manera eficiente.

Formulación Matemática:

La incrustación de vecinos estocásticos distribuidos en t (t-SNE) minimiza la divergencia entre dos distribuciones, una distribución que mide las similitudes por pares de los objetos de entrada y una distribución que mide las similitudes por pares de los puntos de baja dimensión correspondientes en la incrustación. Supongamos que se nos proporciona un conjunto de datos de objetos de entrada $D = \{x_1, x_2, \dots, x_N\}$ y una función $d(x_i, x_j)$ que calcula una distancia entre un par de objetos, por ejemplo, la distancia euclidiana $d(x_i, x_j) = \|x_i - x_j\|$. Nuestro objetivo es aprender una incrustación s -dimensional en la que cada objeto esté representado por un punto, $\varepsilon = \{y_1, y_2, \dots, y_N\}$ con $y_i \in \mathbb{R}^s$ (los valores típicos para s son 2 o 3). Con este fin, t-SNE define probabilidades conjuntas p_{ji} que miden la similitud por pares entre los objetos x_i y x_j simetrizando dos probabilidades condicionales de la siguiente manera:

$$p_{j|i} = \frac{\exp(-d(x_i, x_j)^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-d(x_i, x_k)^2 / 2\sigma_i^2)} \quad (3.62)$$

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2N} \quad (3.63)$$

En la ecuación anterior, el ancho de banda de los núcleos gaussianos, σ_i , se establece de tal manera que la perplejidad de la distribución condicional P_i es igual a una perplejidad predefinida u . Como resultado, el valor óptimo de σ_i varía según el objeto en regiones del espacio de datos con una mayor densidad de datos, σ_i tiende a ser menor que en regiones del espacio de datos con menor densidad. El valor óptimo de σ_i para cada objeto de entrada se puede encontrar mediante una búsqueda binaria simple o utilizando un método robusto de búsqueda de raíces.

En la incrustación s -dimensional ε , las similitudes entre dos puntos y_i e y_j (es decir, los modelos de baja dimensión de x_i y x_j), se miden utilizando un núcleo normalizado de cola pesada. Específicamente, la similitud de incrustación q_{ij} entre los dos puntos y_i e y_j se calcula como un núcleo Student-t normalizado con un solo grado de libertad.

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|y_k - y_l\|^2)^{-1}}, \quad q_{ii} = 0 \quad (3.64)$$

Las colas pesadas del kernel Student-t normalizado permiten modelar objetos de entrada diferentes x_i y x_j mediante contrapartes de baja dimensión y_i e y_j que están demasiado alejadas. Esto crea más espacio para modelar con precisión las pequeñas distancias por pares en la incrustación de baja dimensión. Las ubicaciones de los puntos de incrustación y_i se determinan minimizando la divergencia de Kullback-Leibler entre las distribuciones conjuntas P y Q .

$$c(\varepsilon) = kL(p_{||}Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}} \quad (3.65)$$

Debido a la asimetría de la divergencia Kullback-Leibler, la función objetivo se centra en modelar valores altos de p_{ij} , mediante valores altos de q_{ij} (puntos cercanos en el espacio de incrustación). La función objetivo no es convexa en la incrustación ε . Buscando minimiza descendiendo a lo largo del gradiente:

$$\frac{\partial c}{\partial y_i} = 4 \sum_{j \neq i} (P_{ij} - q_{ij}) q_{ij} Z(y_i - y_j) \quad (3.66)$$

donde definimos el término de normalización $z = \sum_{k \neq l} (1 + \|y_k - y_l\|^2)^{-1}$. Es sencillo ver que la evaluación de las distribuciones conjuntas P y Q es $O(N^2)$, porque ambas distribuciones implican un término de normalización que suma todos los $N(N-1)$ pares de objetos únicos. Dado que t-SNE escala cuadráticamente el número de objetos N, su aplicabilidad se limita a conjuntos de datos con solo unos pocos miles de objetos de entrada, más allá de eso, el aprendizaje se vuelve demasiado lento para ser práctico (y los requisitos de memoria se vuelven demasiado grandes).

3.3.5 LDA (Latent Dirichlet Allocation)

El Análisis de Discursos Latentes (LDA) es un algoritmo de modelado de temas utilizado en procesamiento de lenguaje natural y minería de textos. LDA fue propuesto por David Blei [84]. Su objetivo principal es descubrir temas latentes dentro de un conjunto de documentos y asignar cada documento a una combinación de temas, ver Figura 3.10. A continuación, se describe cómo funciona este algoritmo.

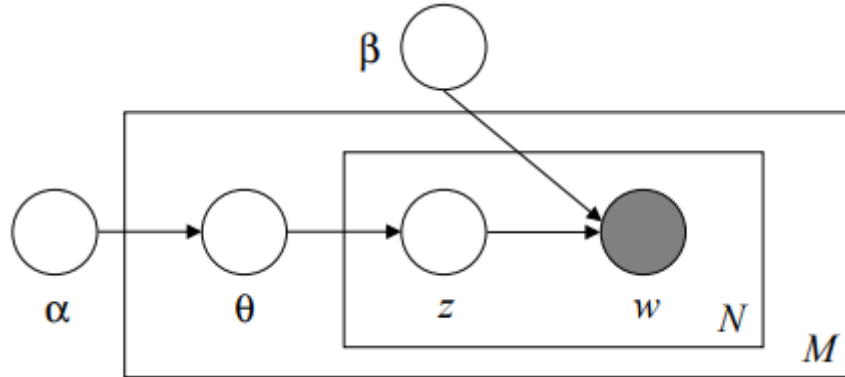


Figura 3.10 LDA Modelo generativo de tres niveles [83].

Inicialización: Se realiza un preprocesamiento en los documentos, que incluye la eliminación de stopwords, tokenización y lematización (opcional), para convertir el texto en una representación más estructurada. El LDA se basa en un modelo generativo que

supone que cada documento se crea a partir de una mezcla de temas y que cada palabra en el documento se elige a partir de una distribución de palabras asociada con un tema. La inferencia estadística se utiliza para retroceder desde los documentos observados hacia la distribución de temas latentes que podría haber generado esos documentos. Una vez que el modelo se ha entrenado, se obtienen las distribuciones de temas para cada documento y las distribuciones de palabras para cada tema. Estas distribuciones permiten interpretar los temas y asignar documentos a temas específicos.

Formulación Matemática:

LDA asume el siguiente proceso generativo para cada documento w en un corpus D :

1. Elija $N \sim \text{Poisson}(\xi)$.
2. Elija $\theta \sim \text{Dir}(\alpha)$.
3. For each of the N words w_n :
 - a. Elija un tema $z_n \sim \text{Multinomial}(\theta)$.
 - b. Elija una palabra w_n de $p(w_n|z_n, \beta)$, una probabilidad multinomial condicionada al tema z_n .

En este modelo básico se hacen varios supuestos simplificadores. Primero, la dimensionalidad k de la distribución de Dirichlet (y por tanto la dimensionalidad de la variable temática z) se supone conocida y fija. En segundo lugar, las probabilidades de palabras están parametrizadas $k \times V$ por una matriz β , donde $\beta_{ij} = P(w^j = 1 | z^i = 1)$, que por ahora tratamos como una cantidad fija eso es para estimar. Finalmente, el supuesto de Poisson no es crítico para nada de lo que sigue y según sea necesario, se pueden utilizar distribuciones de longitud de documentos más realistas. Se debe tener en cuenta que N es independiente de todas las demás variables generadoras de datos (θ y z). Por tanto, es una variable auxiliar y generalmente se ignora su aleatoriedad en el desarrollo posterior.

Una variable aleatoria de Dirichlet k -dimensional θ puede tomar valores en el $(k-1)$ -simplex (un k -vector θ se encuentra en el $(k-1)$ -simplex si $\theta_i \geq 0, \sum_{i=1}^k \theta_i = 1$) y tiene la siguiente densidad de probabilidad en este simplex:

$$p(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^k \alpha_i)}{\prod_{i=1}^k \Gamma(\alpha_i)} \theta_1^{\alpha_1-1} \dots \theta_k^{\alpha_k-1} \quad (3.67)$$

donde el parámetro α es un k -vector con componentes $\alpha_i > 0$, y donde $\Gamma(x)$ es la función Gamma. La Dirichlet es una distribución conveniente en el simplex que pertenece a la familia exponencial, tiene estadísticas suficientes de dimensión finita y es conjugada con la distribución multinomial. Dados los parámetros α y β , la distribución conjunta de una mezcla de temas θ , un conjunto de N temas z y un conjunto de N palabras w se da por:

$$p(\theta, z, w | \alpha, \beta) = p(\theta | \alpha) \prod_{n=1}^N p(z_n | \theta) p(w_n | z_n, \beta) \quad (3.68)$$

donde $p(z_n|\theta)$ es simplemente θ_i para el único i tal que $z_n^i = 1$. Integrando sobre θ y sumando sobre z , obtenemos la distribución marginal de un documento:

$$P(w|\alpha|\beta) = \int p(\theta | \alpha) (\prod_{n=1}^N \sum_{z_n} p(z_n|\theta) p(w_n|z_n, \beta)) d\theta \quad (3.69)$$

Finalmente, tomando el producto de las probabilidades marginales de documentos individuales, obtenemos la probabilidad de un corpus:

$$P(D | \alpha, \beta) = \prod_{d=1}^M \int p(\theta_d | \alpha) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn}|\theta_d) p(w_{dn}|z_{dn}, \beta) \right) d\theta_d \quad (3.70)$$

3.4 Aprendizaje por Refuerzo

El aprendizaje por Refuerzo es una técnica donde el algoritmo aprende y mejora en su respuesta a partir de la experiencia, utilizando un proceso de retroalimentación. El algoritmo aprende por sí solo, obteniendo premios, recompensas y penalizaciones hasta encontrar la forma óptima de ejecutar la tarea. El algoritmo creará la mejor estrategia (política) para obtener una mayor recompensa en tiempo y forma, estas políticas estarán definiendo las acciones que debe de tomar ante cada situación a la que se enfrente. El algoritmo se utiliza comúnmente con este tipo de técnicas.

3.4.1 Q-Learning

El Q-Learning es una técnica de aprendizaje por refuerzo que se utiliza para que un agente aprenda a tomar decisiones secuenciales en un entorno desconocido. A través del tiempo, el agente aprende una función de valor llamada "Q" que estima el valor esperado de tomar una acción particular en un estado específico y seguir una política óptima a partir de ese punto. Q-Learning fue propuesto por Christopher [85]. A continuación, se describe cómo funciona este algoritmo. Se crea una tabla llamada Q-Tabla que contiene valores iniciales para todas las combinaciones de estados y acciones posibles. El agente selecciona una acción para tomar en el estado actual según una política de selección de acción. Esto puede ser mediante una política epsilon-greedy, que equilibra la exploración y la explotación. El agente ejecuta la acción seleccionada en el entorno y observa el nuevo estado y la recompensa asociada. El agente actualiza el valor Q correspondiente al estado y acción actuales utilizando la ecuación de actualización de Q-Learning. Esto implica una combinación ponderada entre el valor Q actual y la recompensa obtenida, además de la estimación del mejor valor Q futuro. El agente pasa al nuevo estado observado y repite el proceso desde el paso 2 hasta que alcanza un estado de finalización o un límite de tiempo. A lo largo del tiempo, la Q-Tabla se actualiza a medida que el agente interactúa más con el entorno, lo que le permite mejorar sus decisiones y aprender una política óptima.

Formulación Matemática:

La convergencia se demuestra construyendo un proceso de decisión de Markov hipotético llamado proceso de acción-repetición, que es similar al proceso real. Luego se muestra que los valores Q producidos por el proceso de Q -learning de un paso después de “ n ” ejemplos de entrenamiento son los valores de acción óptimos exactos para el inicio del proceso de repetición de acciones para “ n ” ejemplos de entrenamiento.

$$Q_n(x, a) = \{(1 - \alpha)Q_{n-1}(x, a) + \alpha[r_n + V_{n-1}(y_n)]\}, \quad (3.71)$$

donde $x = x_n$ y $a = a_n$, $Q_{n-1}(x, a)$ de otra manera $V_n(\cdot)$, es la función de valor actualizada en el estado n .

		Acciones			
		a1	a2	...	an
Estados	X1	Q(X1, a1)	Q(X1, a2)	...	Q(X1, an)
	X2	Q(X2, a1)	Q(X2, a2)	...	Q(X2, an)
	X3	Q(X3, a1)	Q(X3, a2)	...	Q(X3, an)
	...	...	...	...	...
	Xn	Q(Xn, a1)	Q(Xn, a2)		Q(Xn, an)

Figura 3.11 Tabla Q Acciones – Estados.

La forma de demostrar que la función de valor de acción aprendida, Q , se aproxima directamente a Q^* , la función de valor de acción óptima, independientemente de la política que se siga. Aquí, se puede afirmar que demostrar la convergencia para Q -Learning es lo mismo que demostrar que $Q_n(x, a) \rightarrow Q^*(x, n)$ como $n \rightarrow \infty$, ver Figura 3.11. La clave para la prueba de convergencia es un proceso de Markov controlado artificialmente llamado proceso de acción-repetición (ARP). El ARP es un proceso de decisión de Markov puramente teórico, que se utiliza como dispositivo de prueba. Este proceso se construye progresivamente a partir de la secuencia de observaciones.

3.4.2 DDPG (Deep Deterministic Policy Gradient)

El DDPG fue propuesto por Timothy P, et al [86], es un algoritmo de aprendizaje por refuerzo profundo diseñado para abordar problemas de control continuo en entornos de alta dimensión, como el control de robots y juegos en los que las acciones son continuas. El DDPG combina elementos de aprendizaje profundo (redes neuronales) con métodos de gradiente de políticas para aprender políticas determinísticas. A continuación, se describe cómo funciona este algoritmo. Se inicializan dos redes neuronales: una red para aprender la función Q (Q -network) y otra red para aprender la política (actor-network). Estas redes son aproximadoras de funciones que se entrenan para mapear estados a acciones y estimar

los valores Q . El agente interactúa con el entorno y recopila una serie de transiciones (estado, acción, recompensa, nuevo estado) para su uso en el entrenamiento. Se utiliza el error cuadrático medio (MSE) entre las estimaciones Q de la Q -network y las estimaciones de valor objetivo (calculadas a partir de las transiciones recopiladas) para actualizar la Q -network. Se utiliza una técnica llamada "target network" para estabilizar el entrenamiento. La actor-network se entrena para aprender la política determinista óptima. Esto se hace utilizando el gradiente ascendente de la función Q con respecto a las acciones predichas por la actor-network. Periódicamente, se actualizan las redes "target" para seguir las redes principales de Q y actor de manera suave y establecer objetivos más estables. El DDPG utiliza un ruido aditivo en las acciones predichas por la actor-network para fomentar la exploración durante el entrenamiento.

Formulación Matemática:

Tener un objetivo μ' como Q' era necesario para tener objetivos estables y_i para poder entrenar consistentemente al crítico sin divergencias. Esto puede ralentizar el aprendizaje, ya que la red de destino retrasa la propagación de las estimaciones de valor. Sin embargo, en la práctica encontraron que esto fue ampliamente superado por la estabilidad del aprendizaje. El gran desafío del aprendizaje en espacios de acción continua es la exploración. Una ventaja de los algoritmos fuera de políticas como DDPG es que podemos tratar el problema de la exploración independientemente del algoritmo de aprendizaje como se detalla en la Figura 3.12. Construimos una política de exploración μ' agregando ruido muestreado de un proceso de ruido N a nuestra política de actores.

$$\mu'(s_t) = \mu(s_t|\theta_t^\mu) + N \quad (3.72)$$

N se puede elegir para adaptarse al entorno, para generar exploración temporalmente correlacionada para la eficiencia de la exploración en problemas de control físico con inercia.

Algorithm 1 DDPG algorithm

Randomly initialize critic network $Q(s, a|\theta^Q)$ and actor $\mu(s|\theta^\mu)$ with weights θ^Q and θ^μ .
Initialize target network Q' and μ' with weights $\theta^{Q'} \leftarrow \theta^Q$, $\theta^{\mu'} \leftarrow \theta^\mu$.
Initialize replay buffer R
for episode = 1, M **do**
 Initialize a random process $\mathcal{N}$ for action exploration
 Receive initial observation state s_1
 for t = 1, T **do**
 Select action $a_t = \mu(s_t|\theta^\mu) + \mathcal{N}_t$ according to the current policy and exploration noise
 Execute action a_t and observe reward r_t and observe new state s_{t+1}
 Store transition (s_t, a_t, r_t, s_{t+1}) in R
 Sample a random minibatch of N transitions (s_i, a_i, r_i, s_{i+1}) from R
 Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'}))|\theta^{Q'}$
 Update critic by minimizing the loss: $L = \frac{1}{N} \sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$
 Update the actor policy using the sampled policy gradient:

$$\nabla_{\theta^\mu} J \approx \frac{1}{N} \sum_i \nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu} \mu(s|\theta^\mu)|_{s_i}$$

 Update the target networks:

$$\theta^{Q'} \leftarrow \tau \theta^Q + (1 - \tau) \theta^{Q'}$$

$$\theta^{\mu'} \leftarrow \tau \theta^\mu + (1 - \tau) \theta^{\mu'}$$

 end for
end for

Figura 3.12 Algoritmo DDPG [86]

En este algoritmo los autores reconocen que quedan algunas limitaciones a su enfoque propuesto, como ocurre con la mayoría de los enfoques de refuerzo sin modelos, DDPG requiere una gran cantidad de episodios de capacitación para encontrar soluciones.

3.4.3 PPO (Proximal Policy Optimization)

El PPO fue propuesto por John Schulman, et al. [87], es un algoritmo de aprendizaje por refuerzo diseñado para encontrar políticas óptimas en entornos de control secuencial. Utiliza una forma de optimización de políticas que busca mejorar la política actual mientras evita cambios drásticos para garantizar la estabilidad del aprendizaje. A continuación, se describe cómo funciona este algoritmo. Se inicia una política que define la probabilidad de tomar acciones en función de los estados. Esta política puede ser representada por una red neuronal. El agente interactúa con el entorno y recopila un conjunto de datos de experiencias que incluyen estados, acciones, recompensas y probabilidades de acción bajo la política actual. El PPO realiza múltiples iteraciones de optimización de políticas. En cada iteración, se utiliza un método de optimización, como el gradiente conjugado, para encontrar una nueva política que mejore el rendimiento en función de los datos recopilados. Se aplica una restricción para evitar cambios demasiado grandes en la política. La nueva política se evalúa en el entorno para estimar su rendimiento. Esto se hace mediante la recopilación de datos adicionales y la comparación con la política anterior. Si la nueva política es mejor o al menos tan buena como la política

anterior, se acepta. La política actual se actualiza con la nueva política obtenida en la optimización. Si no se cumple esta condición, se rechaza la nueva política. El proceso de recolección de datos, optimización, evaluación y actualización de políticas se repite durante un número específico de iteraciones o hasta que se cumpla un criterio de convergencia.

Formulación Matemática:

Para implementaciones que utilizan diferenciación automática, simplemente se construye la pérdida L^{CLIP} o L^{KLPE} en lugar de L^{PG} , y se realizan múltiples pasos de ascenso de gradiente estocástico en este objetivo. Combinando estos términos, obtenemos el siguiente objetivo, que se maximiza (aproximadamente) en cada iteración:

$$L_t^{CLIP+VF+S}(\theta) = \hat{E}_t[L_t^{CLIP}(\theta) - c_1 L_t^{VF}(\theta) + C_2 S(\pi_\theta)(s_t)] \quad (3.73)$$

Donde c_1 y c_2 son coeficientes, y S detona una bonificación de entropía y L_t^{VF} es una pérdida por error al cuadrado $(V_\theta(s_t) - V_t^{targ})^2$. Un estilo de implementación de gradiente de políticas y muy adecuado para su uso con redes neuronales recurrentes, ejecuta la política durante T pasos de tiempo (donde T es mucho menor que la duración del episodio) y utiliza las muestras recopiladas para actualizar. Este estilo requiere un estimador de ventaja que no mire más allá del paso de tiempo T .

$$\hat{A}_t = -V(s_t) + r_t + \gamma r_{t+1} + \dots + \gamma^{T-t+1} r_{T-1} + \gamma^{T-t} V(s_T) \quad (3.74)$$

donde t especifica el índice de tiempo en $[0, T]$, dentro de un segmento de trayectoria de longitud T determinado. Generalizando esta elección, se usa una versión truncada de la estimación de ventaja generalizada.

$$\hat{A}_t = \delta_t + (\gamma\lambda)\delta_{t+1} + \dots + \dots + (\gamma\lambda)^{T-t+1}\delta_{T-1}, \quad (3.75)$$

donde $\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$. El algoritmo de optimización de políticas proximales (PPO) utiliza segmentos de trayectoria de longitud fija. En cada iteración, cada uno de los N actores (paralelos) recopila T pasos de tiempo de datos. Luego se construye la pérdida sustituta en estos pasos de tiempo de datos NT y la optimizamos con SGD para épocas K , ver Figura 3.13.

Algorithm 1 PPO, Actor-Critic Style

```

for iteration=1, 2, ... do
  for actor=1, 2, ...,  $N$  do
    Run policy  $\pi_{\theta_{old}}$  in environment for  $T$  timesteps
    Compute advantage estimates  $\hat{A}_1, \dots, \hat{A}_T$ 
  end for
  Optimize surrogate  $L$  wrt  $\theta$ , with  $K$  epochs and minibatch size  $M \leq NT$ 
   $\theta_{old} \leftarrow \theta$ 
end for

```

Figura 3.13 Estilo actor-crítico del algoritmo PPO [87].

3.5 Aplicaciones del aprendizaje automático (ML)

El uso de algoritmos de Machine Learning con lleva una serie de ventajas significativas que permiten su utilización en una variedad de aplicaciones:

Automatización de Tareas: Permiten automatizar tareas que de otro modo serían difíciles o consumiría mucho tiempo. Esto incluye tareas de análisis de datos, toma de decisiones y clasificación de información.

Predicciones Precisas: Pueden aprender patrones complejos en los datos y, una vez entrenados, son capaces de hacer predicciones precisas en función de nuevos datos de entrada. Esto es valioso en la toma de decisiones basadas en datos.

Adaptabilidad: Puede adaptarse y mejorar con el tiempo a medida que se les proporcionan más datos y retroalimentación. Esto los hace ideales para aplicaciones que enfrentan cambios constantes.

Descubrimiento de Insights: Revelan patrones y relaciones en los datos que pueden no ser evidentes para los seres humanos. Esto puede llevar al descubrimiento de información valiosa y nuevas perspectivas.

Eficiencia en el Procesamiento de Datos: En aplicaciones que involucran grandes cantidades de datos, pueden procesar y analizar datos a una velocidad mucho mayor que los métodos tradicionales. Esto incluye la visión por computadora, el procesamiento de lenguaje natural, la clasificación y la predicción.

Escalabilidad: Pueden escalar para manejar grandes volúmenes de datos y aplicaciones en tiempo real. Esto es fundamental para empresas y aplicaciones en crecimiento.

Personalización: Permiten la personalización en diversas aplicaciones, desde recomendaciones de productos en línea como Netflix, Amazon y Spotify hasta sistemas de atención médica personalizados para el diagnóstico médico, la detección temprana de enfermedades y la investigación de medicamentos.

Detección de Anomalías: Pueden identificar patrones anómalos o comportamientos inusuales en los datos, lo que es útil para la detección de fraudes, seguridad cibernética donde pueden identificar comportamientos inusuales en grandes conjuntos de datos.

Avances en Investigación Científica: En campos como la biología, química, física y astronomía ayudan a analizar datos y acelerar la investigación científica.

Innovación Empresarial: Impulsa la innovación en empresas de todos los sectores, desde la atención médica y la fabricación hasta la publicidad y el transporte.

3.6 Conclusiones del capítulo

En este capítulo se detallaron algunos de los algoritmos y técnicas de aprendizaje automático que se utilizan, los cuales van evolucionando con el transcurso de los años, para efectos de esta investigación solo mencionamos las tres técnicas más comunes, existen otras como el aprendizaje profundo y más recientemente el aprendizaje cuántico. Cada uno tiene sus propias ventajas y desventajas, y la elección del algoritmo adecuado depende del problema específico que se esté abordando. Además, el campo del aprendizaje automático sigue evolucionando, y se están desarrollando constantemente nuevas técnicas y algoritmos para abordar una amplia variedad de aplicaciones. Sin embargo, también es importante tener en cuenta que el Machine Learning no es una solución que pueda ser utilizada para todos los problemas. Requiere una cantidad significativa de datos de entrenamiento, selección adecuada de algoritmos, sintonización de hiperparámetros y evaluación cuidadosa. Además, debe abordarse con consideraciones éticas, como la privacidad y la equidad, para garantizar que se utilice de manera responsable. En general, el Machine Learning es una herramienta poderosa que tiene el potencial de transformar la forma en que abordamos los problemas y tomamos decisiones en una amplia variedad de campos.

Capítulo 4. ALGORITMOS BIOINSPIRADOS

En este capítulo exploramos aspectos de los algoritmos bioinspirados, que son un tipo de algoritmo de optimización y búsqueda que se inspira en principios y procesos observados en la naturaleza para resolver problemas complejos en campos como la informática, la ingeniería y la ciencia. Estos algoritmos se basan en la idea de que los sistemas naturales han evolucionado para encontrar soluciones efectivas a lo largo de millones de años y, por lo tanto, pueden proporcionar ideas valiosas para el diseño de algoritmos que resuelvan problemas en diversos dominios.

4.1 Inteligencia de Enjambre

La inteligencia de enjambre [88] es un enfoque metaheurístico de optimización. Es un campo de la inteligencia artificial y la computación bioinspirada que se basa en el comportamiento colectivo de sistemas descentralizados y distribuidos para resolver problemas y realizar tareas de manera eficiente. Está inspirada por los comportamientos físicos de los animales que trabajan juntos de manera coordinada y auto organizada para lograr objetivos comunes, como la optimización por enjambres de partículas (PSO, por sus siglas en inglés, Particle Swarm Optimization), la optimización por colonia de hormigas (ACO, por sus siglas en inglés Ant Colony Optimization), y la optimización por colonia o colmena de Abejas (ABC, por sus siglas en inglés Artificial Bee Colony), entre otras. Estas técnicas de optimización han resuelto diferentes tipos de problemas del mundo real como ambientales [89], industriales [90] [91], comerciales [92], de ciudades inteligentes [93] y programación de tareas [94], en búsqueda de la mejor solución global. Este tipo de metaheurísticas siguen 5 principios básicos que son el de proximidad, calidad, respuesta diversa, estabilidad y adaptabilidad.

4.1.1 Optimización por enjambre de partículas

El algoritmo PSO es una metaheurística evolutiva y de búsqueda basado en una metáfora de enjambres de partículas propuesta por Kennedy y Eberhart [95]. Son utilizados para resolver problemas de optimización para buscar el mínimo o el máximo de una función objetivo. El PSO se basa en la generación de un enjambre de partículas que se mueven a través del espacio de búsqueda, representando diferentes soluciones posibles para el

problema. Cada partícula tiene una posición (x) inicial en el espacio de búsqueda, que representa una solución del problema a resolver, y una velocidad (v) que da la dirección y la dimensión del cambio en la posición. Cada partícula debe de ser evaluada calculado el valor de la función objetivo en la posición que ocupa dicha partícula en ese momento, manteniendo una “memoria” de su mejor posición personal hasta el momento “pbest” (P_b), y también se comparte una “memoria” global del mejor resultado encontrado por cualquier partícula en el enjambre “gbest” (G_b) ver Figura 4.1.

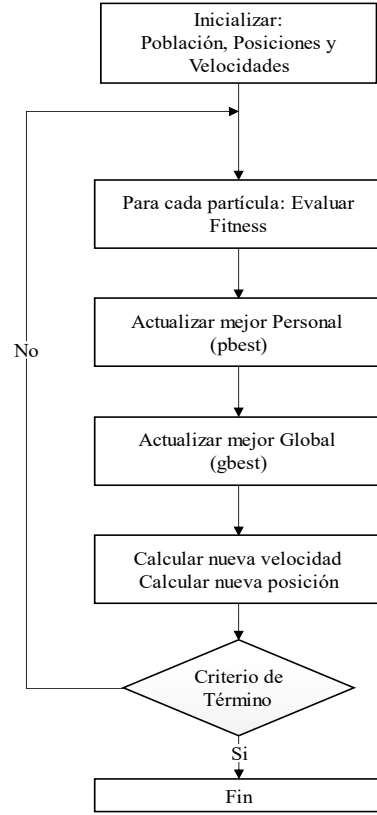


Figura 4.1 Diagrama General del PSO.

La actualización de la velocidad y la posición de una partícula se basa en una combinación ponderada de su velocidad actual lo que permite que las partículas evolucionen hacia soluciones óptimas conforme el algoritmo se sigue repitiendo hasta que se alcanza un criterio de terminación o con el número fijo de iteraciones (t). La mejor solución encontrada por el enjambre se considera como la solución óptima para el problema.

$$v_{ij}^{(t+1)} = \omega v_{ij}^{(t)} + c_1 r_1 \left(P_{bij}^{(t)} - x_{ij}^{(t)} \right) + c_2 r_2 \left(G_{bj}^{(t)} - x_{ij}^{(t)} \right) \quad (4.1)$$

Donde la representación del subíndice (i) es representado desde 1 hasta N número de partículas y (j) se representa desde 1 hasta N dimensiones. El valor del factor utilizado

para (w) representa el peso de la inercia [96] regularmente utilizado entre 0.8 y 1.2, el producto obtenido entre la inercia y la velocidad actual nos da como resultado el *momentum*, los factores de *aceleración* son recibidos en las constantes de $c1$ y $c2$ que suelen ser representados entre un rango de $[0,2]$ así como de $r1$ y $r2$ que son hiperparámetros utilizados para causar perturbaciones aleatorias entre 0 y 1. En la ecuación de velocidad influyen tres partes:

La parte de Inercia $\omega v_{ij}^{(t)}$ que en conjunto con el producto de la velocidad nos da el *momentum* la cual es responsable de mantener la tendencia de la partícula en la dirección en la cual ha estado recorriendo hasta ese momento. Si $\omega < 1$ significa que la partícula va desacelerando conforme avanzan las interacciones, dando como resultado una menor exploración pero una convergencia hacia el óptimo más rápido. Si $\omega > 1$, significa que la partícula se va acelerando, permitiendo una exploración más extensa en las zonas del espacio pero dificultando la convergencia.

La parte cognitiva $c_1 r_1 (P_{(bij)}^{(t)} - x_{ij}^{(t)})$ es la responsable de que la partícula se mueva hacia la posición donde se han obtenido mejores resultados acorde a la experiencia propia de la partícula hasta ese momento. Utilizando el factor de aceleración $c1$ y del parámetro de perturbación aleatoria $r1$ para mejorar la capacidad de no caer en mínimos locales.

La parte social $c_2 r_2 (G_{bj}^{(t)} - x_{ij}^{(t)})$ es la responsable de que la partícula se mueva hacia la mejor posición global con el intercambio de información y el comportamiento social como enjambre encontrado hasta el momento. Utilizando el factor de aceleración $c2$ y del parámetro de perturbación aleatoria $r2$ para mejorar la capacidad de no caer en mínimos locales.

La posición de la partícula es actualizada con la velocidad encontrada con la ecuación anterior y es obtenida con la siguiente ecuación detallada en la Figura 4.2.

$$x_{ij}^{(t+1)} = x_{ij}^{(t)} + v_{ij}^{(t+1)} \quad (4.2)$$

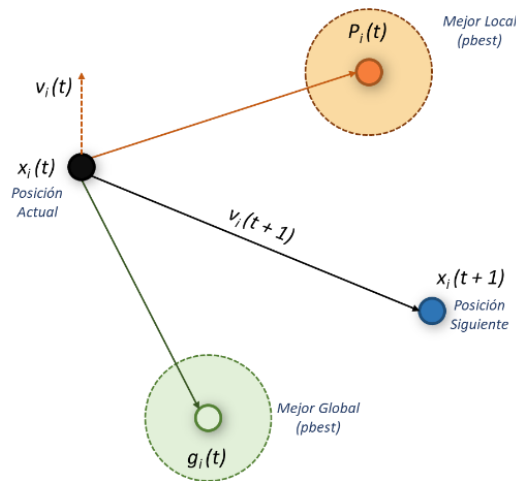


Figura 4.2 Actualización de la posición de la i-th partícula.

4.1.2 Optimización de colonia de hormigas

El algoritmo de optimización de colonia de hormigas (ACO) fue propuesto por primera vez por Marco Dorigo [97]. Es un método inspirado por el comportamiento de colonias de algunas especies de hormigas, por una forma de comunicación conocida como estigmergia (feromona). Esto se debe a que, mientras las hormigas van desde la colonia a las fuentes de alimento y viceversa, van dejando un rastro de feromonas. Ya que las hormigas que pasan por caminos más cortos tardarán menos tiempo en moverse entre el recurso y la colonia, este camino tendrá un rastro de feromonas mayor al pasar más veces las hormigas por él. De esta forma las hormigas consiguen ir siempre por el camino más corto hasta un recurso, ya que las otras hormigas de la colonia siguen los que presentan un rastro con feromonas más concentradas

El algoritmo ACO imita el comportamiento de las colonias de hormigas. Cada $arc(i, j)$ del gráfico $G = (N, A)$ lo asociamos a una variable τ_{ij} llamada rastro de feromona artificial. Las hormigas leen y escriben los rastros de feromonas. La cantidad (intensidad) de un rastro de feromonas es proporcional a la utilidad estimada por las hormigas de usar ese arco para construir buenas soluciones. Cada hormiga construye, a partir del nodo fuente, una solución al problema aplicando una política de decisión paso a paso. En cada nodo, la hormiga lee (detecta) la información local almacenada en el propio nodo o en sus arcos salientes y la utiliza de forma estocástica para decidir a qué nodo moverse a continuación. Al comienzo del proceso de búsqueda, se asigna una cantidad constante de feromona (por ejemplo $\tau_{ij} = 1, \forall (i, j) \in A$) a todos los arcos. Cuando se ubica en un nodo i , una hormiga k usa los rastros de feromonas τ_{ij} para calcular la probabilidad de elegir j como siguiente nodo:

$$p_{ij}^k = \begin{cases} \frac{(\tau_{ij})^\alpha (n_{ij})^\beta}{\sum_{j \in \mathcal{N}_i^k} (\tau_{ij})^\alpha (n_{ij})^\beta}, & \text{Si } j \in \mathcal{N}_i^k \\ 0, & \text{Si } j \notin \mathcal{N}_i^k \end{cases} \quad (4.3)$$

$\mathcal{N}_i^k$ es la vecindad de la hormiga k cuando está en el nodo i . En ACO, la vecindad de un nodo i contiene todos los nodos directamente conectados al nodo i en el gráfico $G = (N, A)$, excepto el predecesor del nodo i (es decir, el último nodo que visitó la hormiga antes de pasar a i). De esta forma las hormigas eviten regresar al mismo nodo que visitaron inmediatamente antes del nodo i . Solo en caso de que $\mathcal{N}_i^k$ esté vacío, lo que corresponde a un callejón sin salida en el gráfico, el predecesor del nodo i se incluye en $\mathcal{N}_i^k$. Tenga en cuenta que esta política de decisión puede conducir fácilmente a bucles en las rutas generadas, ver Figura 4.3.

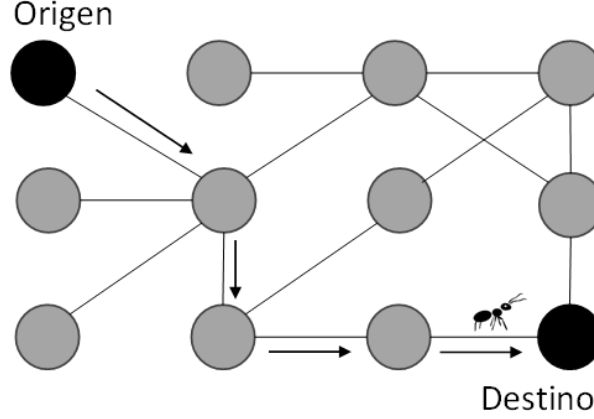


Figura 4.3 Hormigas generando Soluciones de nodo origen a nodo destino.

τ_{ij} es el valor del rastro de feromonas en la arista del nodo i al nodo j , $n_{ij} = 1/d_{ij}$ es un valor heurístico que está disponible a priori, d_{ij} representa la “distancia” que existe entre dos nodos, α y β son dos parámetros (α con valor de $[0,1]$ y $\beta > 0$) que determinan la influencia relativa del rastro de feromonas y la información heurística respectivamente, y $\mathcal{N}_i^k$ es el nodo factible de la hormiga k cuando se encuentra en el nodo i , es decir, el conjunto de nodos que la hormiga k aún no ha visitado. Una vez que todas las hormigas hayan creado su propia solución, es necesario efectuar el depósito de feromonas por los caminos que han recorrido (conexión de nodos) con su respectiva evaporación cumpliendo con los principios del algoritmo $\tau_{ij} = (1 - p)\tau_{ij}$, donde $0 < p < 1$ es la velocidad de evaporación de la feromona. El hiperparámetro p es utilizado para evitar la acumulación ilimitada de los rastros de feromona y permite que el ACO “remueva” las malas decisiones tomadas previamente por las hormigas. Después de la evaporación las hormigas depositan feromonas en las aristas que han cruzado en su recorrido.

$$\tau_{ij} = \tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k \quad (4.4)$$

Donde $\Delta\tau_{ij}^k$ es la cantidad de feromonas depositadas de la hormiga k en las aristas que ha visitado. $\Delta\tau_{ij}^k = \begin{cases} \frac{1}{c^k} & \text{Si la arista } (i, j) \text{ pertenece a } T^k \\ 0 & \text{en caso contrario.} \end{cases}$ Donde c^k es la distancia del recorrido T^k , construida por la k -ésima hormiga y se calcula como la suma de las longitudes de las aristas a las que pertenece a T^k . Con esto, la hormiga que contiene la mejor solución es aquella que ha recorrido la menor distancia entre los nodos.

4.2 Algoritmos Evolutivos

Los algoritmos evolutivos son una de las técnicas metaheurística de optimización y búsqueda de soluciones basados en los postulados de la evolución biológica. En ellos se

mantiene un conjunto de entidades que representan posibles soluciones, las cuales se mezclan, y compiten entre sí, de tal manera que las más aptas son capaces de prevalecer a lo largo del tiempo, evolucionando hacia mejores soluciones cada vez. Los algoritmos evolutivos, y la computación evolutiva, son una rama de la inteligencia artificial. Son utilizados principalmente en problemas con espacios de búsqueda extensos y no lineales, en donde otros métodos no son capaces de encontrar soluciones en un tiempo razonable.

4.2.1 Algoritmos Genéticos

Los algoritmos genéticos (AG) fueron ideados por John Holland [98] en los años 70 y popularizados por uno de sus estudiantes, David Goldberg. El Algoritmo genético es un método probabilístico genérico de búsqueda y optimización basado en evolucionar un conjunto inicial de soluciones candidatas, denominado población de individuos, hacia mejores soluciones a lo largo de generaciones de poblaciones mediante mecanismos bioinspirados como la herencia, la selección natural, la adaptación al medio, el cruzamiento y la mutación. Los métodos de codificación de caracteres, binaria y de valores decimales son los más utilizados por la comunidad científica [98], siendo este último el que se adapta concretamente a la solución de problema de optimización, con la distribución del número de procesos y la cantidad de los tweets que se necesitan balancear y procesar en las siguientes secciones. Un AG está compuesta por la calibración o configuración, los procesos de evolución, validación y adaptación, ver Figura 4.4. Cada uno de los flujos del GA serán detallados en las siguientes secciones.

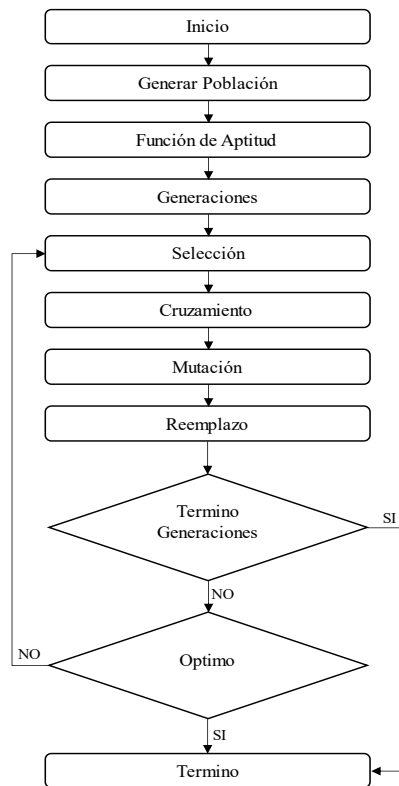


Figura 4.4 Diagrama General de Algoritmo Genético.

Calibración del Algoritmo Genético.

Se calibra el AG con el tamaño de la población, generaciones, probabilidad de mutación, función de costo y tamaño total del porcentaje de la función de costo por generación, entre otros.

Codificación (Genotipo).

Se utilizó una codificación decimal en la forma $(x_1, x_2, x_3, \dots, x_n)$ donde cada variable x_i puede asumir valores reales o enteros dentro de un intervalo permitido (m_i, M_i) , siendo esta una representación genética (genotipo) de las soluciones, con cada cromosoma como un vector numérico que representa una secuencia de tareas. Una opción es codificar los cromosomas mediante permutaciones decimales.

Población.

La creación de la población inicial de individuos (cromosomas), se realiza generalmente con un enfoque aleatorio utilizado en [99] [100] que garantiza la probabilidad de que las soluciones candidatas sean considerablemente diferentes entre sí, lo que ayuda a evitar la convergencia del AG hacia óptimos locales. Cada individuo i , se formará con una permutación de entrada hasta la longitud de genotipo, repitiendo este método hasta que se llegue al tamaño configurado de la población.

Función de Aptitud.

La Función de aptitud o costo [99] [100] es el componente más importante en cualquier método de optimización, ya que evalúa la calidad de las soluciones para determinar su efectividad. Esta función suele adaptarse al problema específico que se desea resolver, lo que significa que varían según las expectativas del algoritmo. En general buscan maximizar el valor, aunque en algunos casos también pueden minimizarlo.

Selección.

El objetivo principal de los operadores de selección es conservar preferentemente las mejores soluciones, es decir, los individuos con mayor fitness. Como ya hemos visto anteriormente es necesario hacer una selección con los individuos más capacitados para que éstos sean los que se reproduzcan con más probabilidad de acuerdo con la teoría de Darwin, en la cual los más capacitados son los que deben sobrevivir y crear una nueva descendencia más facultada. Por lo tanto una vez evaluado cada cromosoma y obtenida su puntuación, se tiene que crear la nueva población teniendo en cuenta que los buenos rasgos de los mejores se transmitan a ésta. Esta selección se puede realizar de varias formas como se detalla a continuación.

- Selección por Rueda de Ruleta: Se crea para esta selección una ruleta [98], y con los cromosomas presentes en una generación. Cada cromosoma tendrá una parte de esa ruleta mayor o menor en función a la puntuación que tenga cada uno. Se hace girar la ruleta y se selecciona el cromosoma en el que se para la ruleta. Obviamente el cromosoma con mayor puntuación saldrá con mayor probabilidad. En caso de que las probabilidades difieran mucho, este método de selección dará problemas puesto que, si un cromosoma tiene un 90% de posibilidades de ser seleccionado, el resto apenas saldrá lo que reduciría la diversidad genética.
- Selección por Rango: En este método a cada cromosoma se le asigna un rango numérico basado en su aptitud y la selección se realiza en base a este ranking.
- Selección Elitista : Este método de selección copia el mejor cromosoma o alguno de los mejores en la nueva población. El elitismo puede mejorar el funcionamiento de los algoritmos genéticos al evitar que se pierda la mejor solución. Una variación del elitismo es que el mejor cromosoma solo se copie a la siguiente generación en caso de que tras una reproducción/mutación no se haya generado un cromosoma mejor.
- Selección por Torneo : Se escogen de forma aleatoria un número de individuos de la población, y el que tiene puntuación mayor se reproduce, sustituyendo su descendencia al que tiene menor puntuación.

Cruzamiento.

El operador de cruzamiento se utiliza para intercambiar genes de los cromosomas seleccionados en la fase anterior, y esto generará una nueva descendencia. Este método puede utilizar varias técnicas como:

- Cruzamiento de un punto (One Point Crossover) [98] [100] [102]: Establece un punto de intercambio en un lugar aleatorio del cromosoma de los dos individuos seleccionados, intercambiando todo su código anterior a ese punto y otro individuo intercambia a partir de ese punto, de esta forma se genera la nueva descendencia.
- Cruzamiento de dos puntos (Two Point Crossover) [98]: Establece el intercambio de los genes que aparecen en el intervalo de genes delimitados por dos puntos seleccionados aleatoriamente.
- Cruzamiento uniforme (Uniform Crossover UC) [103]: Establece el valor de una posición dada en el cromosoma de la descendencia corresponda al valor en la posición del cromosoma de uno de los padres o el valor de esa posición del cromosoma del otro padre.
- Cruzamiento en ciclos (Cycle Crossover CX) [104]: Establece que se crea una descendencia a partir de los padres, de tal manera que cada posición se ocupa por el correspondiente elemento de uno de los padres.

En la literatura la técnica más utilizada a las necesidades del problema es el cruzamiento en un punto, pero para problemas del tipo de planificación de procesos, este tipo de cruzamiento puede tener una desventaja, ya que puede lograr que un mismo proceso se pueda asignar más de una vez y en algunos casos ni sea asignada, por lo cual se determinó utilizar la técnica de cruzamiento en ciclos, que utilizan distribuciones probabilísticas que permiten con mayor frecuencia crear descendientes cercanos a los padres y con poca frecuencia descendientes lejanos a los padres y por lo tanto permite mantener la diversidad genética entre la población.

Ejemplo del cruzamiento.

Primeramente, utilizamos una tupla de padres X e Y , que la técnica de selección nos proporcionó en la Tabla 4.1.

Tabla 4.1 Tupla de Padres seleccionados.

<i>Padre X</i>	1	2	3	4	5	6	7	8	9	10	11	12
<i>Padre Y</i>	9	7	1	4	6	11	2	12	5	3	8	10

Para iniciar esta técnica de cruzamiento en ciclos, debemos escoger un padre que servirá como punto de partida, en nuestro caso seleccionaremos el Padre X y buscaremos encontrar los ciclos que se detectan de la primera posición del Padre X en el Padre Y . siguiendo los siguientes pasos ver Tabla 4.2 y Tabla 4.3.

1. Iniciar el ciclo de alelos.
 - a. Comenzar con el primer alelo del Padre X .
 - b. Mirar el Alelo que está en la posición del Padre Y .
 - c. Ir a la posición con el mismo Alelo del Padre X .

- d. Añadir el Alelo en el ciclo.
- e. Repetir el proceso del b al d.
2. Agregar los alelos del ciclo en la descendencia que tenía el primer padre
3. Repetir el proceso con el Padre Y.

Tabla 4.2 Descendiente del padre X.

<i>Padre X</i>	1	2	3	4	5	6	7	8	9	10	11	12
<i>Padre Y</i>	9	7	1	4	6	11	2	12	5	3	8	10

Donde los ciclos localizados fueron 1 – 9 – 5 – 6 – 11 – 8 – 12 – 10 – 3 – 1

Tabla 4.3 Descendiente del padre Y.

<i>Padre X</i>	1	2	3	4	5	6	7	8	9	10	11	12
<i>Padre Y</i>	9	7	1	4	6	11	2	12	5	3	8	10

Donde los ciclos localizados fueron 9 – 1 – 3 – 10 – 12 – 8 – 11 – 6 – 5 – 9

De esta forma obtuvimos la nueva descendencia de los Hijos X e Y, que serán parte de la nueva población como se puede observar en la Tabla 4.4.

Tabla 4.4 Descendencia de los padres X y Y.

<i>Descendiente X</i>	1	7	3	4	5	6	2	8	9	10	11	12
<i>Descendiente Y</i>	9	2	1	4	6	11	7	12	5	3	8	10

Mutación.

El operador de mutación ayuda a la diversidad genética, y puede prevenir que las soluciones de la población se vean limitadas a llegar a un óptimo local, este operador tiene la finalidad de producir cambios espontáneos en la población de cromosomas lo que agrega una variabilidad extra en los espacios de la solución. El factor de mutación es controlado por una tasa de mutación estática pm , por cada gen del cromosoma de la población actual generamos un número aleatorio m . Si $m < pm$ entonces el gen muto e invertimos el valor del gene (alelo) a otra posición del cromosoma. Tomando como ejemplo los descendientes logrados en el cruzamiento, podemos reflejar como la mutación es aplicada en uno de los descendientes. Donde el valor aleatorio para $m = 0.001$ y $pm = 0.01$ y el alelo encontrado fue en la posición 10, se hace el intercambio del valor del gen entre la posición 1 y 10, como se pude ver en la Tabla 4.5.

Tabla 4.5 Mutación del Descendiente X.

<i>Descendiente X</i>	1	7	3	4	5	6	2	8	9	10	11	12
<i>Descendiente X Mutado</i>	1	11	3	4	5	6	2	8	9	10	7	12

Validación y Reemplazo.

El operador de reemplazo es utilizado para sustituir los mejores cromosomas que surgieron del cruzamiento y la mutación de la población inicialmente procesada, dicho reemplazo se realizó de una forma elitista, preservando a las mejores soluciones a lo largo de las generaciones. El ciclo que se realizó con los diferentes operadores se debe de estar repitiendo acorde a al número de generaciones que se hubieran configurado, el cual se estará repitiendo del operador de selección al operador de reemplazo, hasta que se encuentre el cromosoma más óptimo, o hasta que las generaciones terminen, esta última opción es regularmente la más utilizada, ya que al no conocer el óptimo a buscar se tiene que completar el término de las generaciones.

4.3 Conclusiones del capítulo

En este capítulo se detallaron los algoritmos bioinspirados, que suponen pasar de intentar resolver problemas de optimización de una manera tradicional, que requiere mayor comprensión y no se puede aplicar a problemas más complejos, a “simplemente” copiar un modelo de optimización de algo que ha estado resolviendo este tipo de problemas durante siglos sin ningún conocimiento previo y de forma más eficiente, como la propia la naturaleza. Desde sus inicios han demostrado su potencial para solucionar este tipo de problemas, y con el paso del tiempo van surgiendo cada vez más modelos que se pueden aplicar con mayor facilidad y en más áreas como seguridad informática, ecología, industrial, comercio y ciudades inteligentes por mencionar algunas. Debido a su simplicidad y reducción del procesamiento computacional requerido, han permitido que estas metaheurísticas tomen una importante popularidad en el campo científico, y que en un futuro próximo, este tipo de modelos serán cada vez más utilizados para tomar el control de alguna parte del negocio de empresas, ya sea en la logística de transporte, distribución, etc., aumentando la optimización de resultados, y los beneficios de inversión de las empresas. Dado que este es un campo de investigación relativamente joven, todavía hay mucho espacio para el desarrollo y la aparición de nuevos algoritmos bioinspirados, porque como se mencionó basar la inspiración en la naturaleza a menudo da como resultado algoritmos que son adaptables y robustos en entornos cambiantes o ruidosos. Estas metaheurísticas ofrecen herramientas valiosas para abordar problemas complejos y desafiantes. A pesar de sus ventajas, los algoritmos bioinspirados no son una solución universal y pueden tener limitaciones en determinados contextos. Elegir el algoritmo y la configuración de parámetros correctos es fundamental para su éxito. Además, la interpretación de los resultados a veces puede resultar complicada debido a la falta de transparencia de algunos algoritmos.

Capítulo 5. MATERIALES Y MÉTODOS

En este capítulo se describe la implementación detallada del modelo propuesto en esta investigación para optimizar los tiempos de preprocesamiento en la detección del suicidio en redes sociales. Se especifican los materiales utilizados en el transcurso del desarrollo de la investigación, como lo es el lenguaje de programación con las librerías utilizadas, así como los algoritmos utilizados y finalmente un detalle específico de las etapas utilizadas en nuestro modelo.

5.1 Métodos experimentales

Los métodos experimentales son esenciales para evaluar, comparar y buscar mejorar los modelos propuestos en cualquier investigación. La evaluación del modelo es de gran importancia para poder determinar la precisión, rendimiento, eficiencia y otras métricas relevantes del comportamiento del modelo en situaciones variadas. Otra técnica utilizada es hacer comparaciones entre varios algoritmos para poder determinar la eficiencia entre ellos y bajo ciertas condiciones. La calibración de los hiperparámetros es determinante para encontrar los valores óptimos a utilizar en cualquiera de los modelos seleccionados, esta última estrategia fue utilizada y validada con los siguientes algoritmos que fueron utilizados a lo largo de la investigación.

Algoritmo Genético

El AG se utiliza para resolver problemas de búsqueda y optimización basado en evolucionar un conjunto inicial de soluciones candidatas, este fue calibrado con diferentes hiperparámetros como la población, generaciones, probabilidad de mutación así como estrategias de selección, cruzamiento y mutación, detallado en el Capítulo 4.

Algoritmo PSO

El PSO se basa en la generación de un enjambre de partículas que se mueven a través del espacio de búsqueda, representando diferentes soluciones posibles para el problema, este fue calibrado con diferentes hiperparámetros como número de partículas, iteraciones, factores de aceleración, perturbaciones aleatorias, detallado en el Capítulo 4.

Algoritmo ACO

El ACO es un método inspirado por el comportamiento de colonias de algunas especies de hormigas, por una forma de comunicación conocida como estigmergia, este fue calibrado con diferentes hiperparámetros como número de hormigas, iteraciones, nivel de atracción, nivel de feromonas y grado de evaporación de feromonas, detallado en el Capítulo 4.

Algoritmo Round Robin (RR)

El Algoritmo Round Robin, sigue una distribución cíclica asignando un tiempo "quantum" o "time slice" a cada proceso. En sistemas con núcleos homogéneos, donde todos los núcleos tienen capacidades similares, esta distribución es adecuada en términos de asignación de tiempo de CPU, ya que asegura que cada proceso reciba una cantidad igual de tiempo de ejecución. Sin embargo, el tamaño de los requerimientos de las tareas no es homogéneo. Esto significa que, aunque la distribución del tiempo de CPU sea cíclica, no garantiza una distribución equilibrada de la carga de trabajo. Las tareas con diferentes necesidades de recursos pueden llevar a un desequilibrio en la carga entre los núcleos. Este desequilibrio puede afectar el rendimiento y la eficiencia del sistema, ya que algunos núcleos pueden estar poco cargados mientras otros están sobrecargados.

Algoritmo First-Come, First-Served (FCFS)

El Algoritmo First-Come, First-Served, gestiona los procesos en el orden en que llegan al sistema. En sistemas con núcleos homogéneos, esta distribución se realiza de manera que el proceso que llega primero es el primero en ser ejecutado. Sin embargo, este enfoque puede causar el problema conocido como "convoy effect", donde procesos cortos pueden esperar un tiempo considerable si hay procesos largos en la cola. Esto ocurre debido a que los requerimientos de recursos de las tareas no son homogéneos. En FCFS, una tarea de larga duración puede bloquear la ejecución de tareas más cortas que llegaron posteriormente, causando ineficiencias en el sistema. Como resultado, los núcleos pueden experimentar cargas de trabajo desiguales, lo que puede llevar a ineficiencias en el rendimiento del sistema.

Algoritmo Shortest Job First (SJF)

El Algoritmo Shortest Job First, selecciona el proceso con el menor tiempo de ejecución esperado. En sistemas con núcleos homogéneos, esta distribución es adecuada en términos de asignación de tiempo de CPU, ya que todos los núcleos tienen capacidades similares. Sin embargo, la variabilidad en los requerimientos de recursos de las tareas puede llevar a desequilibrios, puesto que las tareas no son homogéneas. Esto puede resultar en situaciones donde algunos núcleos se encuentren sobrecargados con tareas más pesadas después de haber procesado varias tareas cortas, mientras que otros núcleos estén poco

cargados. Esta utilización desigual de los recursos del sistema puede afectar negativamente el rendimiento y la eficiencia.

K-Nearest Neighbors (K-NN)

El algoritmo clasifica cada dato nuevo en el grupo que corresponda, según los “K” vecinos más cerca de un grupo o de otro. La implementación fue calibrada con 3, 5 y 7 vecinos (K), utilizando este parámetro para calcular la distancia del nuevo elemento a cada uno de los existentes y ordenar la distancia de menor a mayor para seleccionar el grupo al que pertenece, logrando obtener los mejores resultados $k=5$, detallado en el Capítulo 3.

Logistic Regression

Este algoritmo es utilizado para predecir el resultado de una variable categórica. Se experimentó calibrando de forma multinomial con “lbfgs”, “saga”, y “sag” al solucionador del algoritmo, obteniendo mejores resultados con la última configuración, detallado en el Capítulo 3.

Support Vector Machine (SVM)

Este algoritmo generalmente trabaja sobre el hiperplano ideal que es más útil para clasificar nuevas ilustraciones. Se experimentó calibrando el kernel con las funciones sigmoid y lineal obteniendo mejores resultados con la última configuración, detallado en el Capítulo 3.

Random Forest

Este algoritmo considera numerosos árboles de decisión formando así un bosque, tratando de alcanzar la mejor característica al azar. Se experimentó calibrándolo inicialmente 100, 150 y 200 el número de árboles en el bosque (estimators), logrando obtener los mejores resultados con la última configuración de 200 árboles, detallado en el Capítulo 3.

Multi-layer Perceptron (MLP)

Este algoritmo es una red neuronal artificial formada por múltiples capas clasificadas en tres tipos entrada, oculta y salida, este es utilizado para resolver problemas que no son linealmente separables. Se experimentó con la red neuronal de feedforward la cual requiere de varios ajustes en los parámetros a utilizar, logrando los mejores resultados con la siguiente calibración, un solucionador del tipo “lbfgs”, una interacción máxima de 2000, 20 capas ocultas, y una razón de aprendizaje de 0.001, detallado en el Capítulo 3.

5.2 Metodología propuesta

El principal objetivo es la optimización del tiempo en la detección de características suicidas, utilizando algoritmos bioinspirados para maximizar los recursos actuales del sistema de cómputo con una adecuada planificación de tareas (tweets). La metodología plantea diferentes etapas relevantes que se emplearon para la construcción de nuestro modelo que serán detalladas a continuación, ver Figura 5.1.

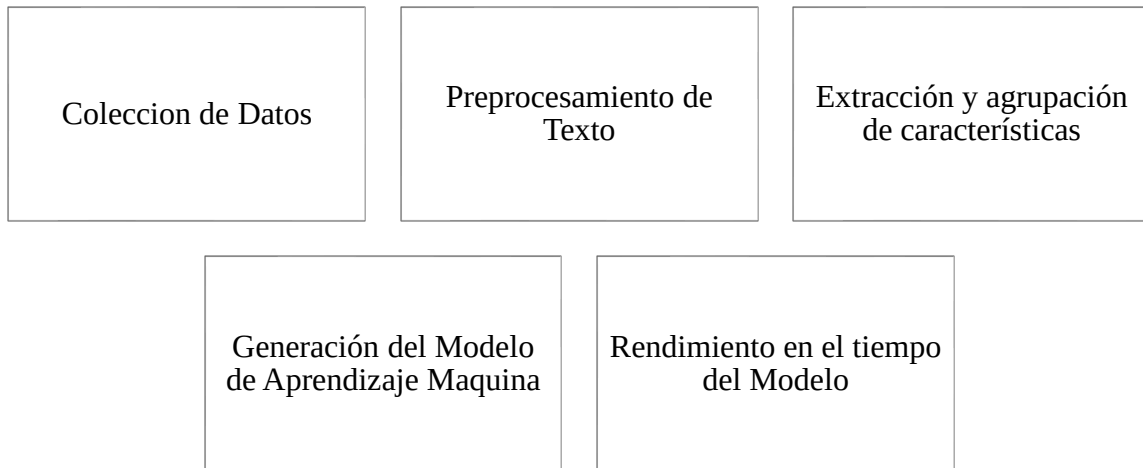


Figura 5.1 Metodología propuesta.

5.2.1 Formulación del problema

El problema de la programación de tareas consiste en encontrar la asignación óptima de un número de tareas a un número determinado de procesadores. Para modelar este problema, se consideran los siguientes supuestos:

- Todas las tareas enviadas son independientes
- Una tarea no puede ser asignada a más de un procesador.

5.2.2 Sistema de Computo

El sistema está dividido en dos principales arquitecturas: la física y la lógica. La arquitectura física está compuesta por un sistema hyperthreading Intel(R) Xeon(R) E-2176M CPU @ 2.70GHz, 2712 MHz con múltiples procesadores $px \in P$ ($x = 1, 2, 3, \dots, Y_p$) que son heterogéneos en su núcleo y velocidad de computación. Cada CPU puede procesar los tweets con los recursos que el sistema le pueda proporcionar tanto de hardware y/o software, tales como programas, bibliotecas, monitoreo de memoria, etc. El sistema necesita ser configurado de tal forma que se pueda tener un límite máximo de

utilización de los recursos computacionales $U_{px}^{max} \in RC$ ($0 < U_{px}^{max} < 100$) para no afectar el sistema de cómputo en general y permita la correcta distribución de cargas internas del sistema operativo utilizado, el cual se describió en el apartado de Materiales. La arquitectura lógica estará compuesta por el software utilizado, el sistema descrito en el apartado de Materiales, con el cual se desarrollaron diferentes métodos como, la recolección de datos de Twitter, la distribución de procesos con el tamaño del número tweets, el algoritmo para planificar tareas, preprocesamiento de datos, extracción de características, agrupamiento de los datos, generación y validación del modelo.

5.2.3 Colección de Datos

La colección de datos consiste en recopilar tweets relacionados con temas del suicidio en la red social Twitter para crear un conjunto de datos necesario para la distribución, clasificación y predicción mediante técnicas de inteligencia artificial seleccionadas. El filtro de vocabulario es una de las funciones más relevantes de esta sección, ya que se descarta el contenido que no esté relacionado con el suicidio. Estas agrupaciones de vocabulario se extraen de la literatura en contribuciones realizadas por las ciencias de la computación y psicología, como se observa en la Tabla 5.1.

Tabla 5.1 Vocabulario del suicidio en inglés y español.

Vocabulario (Palabras y frases)	
Inglés	
Sueki, Hajime, et al. [105]	Want to die, want to commit suicide
Bridianne O'Dea et al. [106]	I want to die, I want to kill myself, I might just kill myself, when you call me that name, it makes me want to kill myself, I am going to kill myself, if this happens, I will kill myself.
James G. Phillips, et al. [107]	suicide, suicide attempt, crowd, audience, webcam, skype, and livestream.
Cheryl A King, et al. [108]	homicide, suicide risk
S.Y. Lee, et al. [109]	cyber-suicide pacts, suicide pact, contact me for counseling, think of others who love you, the hard times will be gone soon.
Robert A. Fahey, et al. [110]	anger, confusion, disgust, fear, happiness, sadness, shame, surprise
Thomas Niederkrotenthaler, et al. [111]	bottle, glass, cutting, bleeding, self-inflicted, self-inflicted
Marouane Birjali, et al. [8]	fear, depression, harassment
Pete Burnap, et al. [112]	asleep and never wake, do not want to try anymore, end it all, is not worth living, my life is pointless, kill myself, to live any more, want to end it, want to disappear, want to die.
Bart Desmet, et al. [113]	Cutting myself, strongly concerning, possibly concerning or safe to ignore.

Diniz EJS, et al. [10]	suicidal, suicide, kill myself, my suicide note, my suicide letter, end my life, never wake up, can't go on, not worth living, ready to jump, sleep forever, want to die, be dead, better off without me, better of dead, suicide plan, suicide pact, tired of living, don't want to be here, die alone, go to sleep forever.
Fonseca-Pedrero, E. et al. [16]	Going to sleep and wishing you would not get up, felt that life is not worth living
Homan, S., et al. [19]	die, suicide, I might just kill myself, distress, tired living, hate tired, I am afraid, never again
de Oliveira, A. C., et al. [114]	suicide, I want to die, my suicide, desire/wish, I am going to kill, crisis, sadness, I woke up sad. I want to kill myself, what a desire to kill myself
Español	
Casado Blanco, et al. [115]	He dejado de ser yo, estoy cansada de experimentos, estoy trastornada lo malo es que me doy cuenta y sufro, estoy muy chalada, mucho, no me siento entendida, te vas viendo cada día peor diluyéndote, perdiéndote, dejadme descansar, salir de esto que me ata
Saray Zafra Cremades, et al. [116].	La vida es un vacío constante, lleno de soledad y angustia, Tengo una soledad tan concurrida, tan llena de rostros de vos.
Sánchez-Loyo Luis Miguel, et al. [17]	Sin ganas de hacer nada, sin ganas de ver a nadie, no quiere salir, no encuentra la salida, tristeza, suicidio, desesperación, ansiedad, alejarse, estar solo, suicidarse

Con los datos recolectados se generó una lista entre palabras y frases con intención suicida, lo que nos ayudó a acotar la búsqueda y extracción de Twitter a través de la interfaz de programación de aplicaciones (API). Algunos ejemplos: “suicidio, depresión, ansiedad, estrés, tristeza, nota de suicidio, carta de suicidio, terminar con la vida, no despertar nunca, no puedo seguir, no vale la pena vivir la vida, listo para saltar, dormir para siempre, quiero morir, morir...”. Esta recopilación de datos se llevó a cabo en intervalos de aproximadamente 2 horas al día durante 6 meses, lo que resultó en la obtención de más de 1 millón de tweets.

5.2.4 Preprocesamiento de Texto

Este paso es importante en el aprendizaje automático porque las técnicas de PLN tienen como objetivo transformar los datos obtenidos de una representación no estructurada a una representación estructurada a partir de los tweets extraídos en la sección anterior, que deben procesarse de manera eficiente para reducir la ambigüedad y el ruido que pueda haber en el conjunto de datos. Las técnicas de eliminación de ruido en tweets (TER-TWS) realizadas en esta sección, son una recolección de las principales aportaciones realizadas por la literatura [113], [25], [117], [118], [119]. Logrando una diferencia al utilizar el TER-TWS para lograr una “limpia” al contenido, que nos dio buenos resultados con los

algoritmos de ML que utilizamos, y que se explicarán en las siguientes secciones. El TER-TWS está compuesto de la siguiente lista de técnicas.

- Reemplazar abreviaciones [don't].
- Remover viejo estilo de retweets (RT).
- Cambiar todo el texto del comentario a minúsculas.
- Reemplazar las URL de sitios web por variable local.
- Reemplazar @username por variable local.
- Reemplazar espacios en blanco.
- Reemplazar Hashtag (#).
- Eliminar caracteres especiales y números.
- Eliminar correos electrónicos.
- Eliminar saltos de línea.
- Remover palabras duplicadas.
- Aplicar la técnica Stop Words o palabras vacías.

Esta etapa requiere altos costos computacionales. y es aquí donde cobra relevancia nuestra aportación en forma de optimización de los recursos del sistema (procesadores) mediante los algoritmos bioinspirados.

5.2.5 Extracción y agrupación de características

Los algoritmos de aprendizaje automático no pueden procesar el texto en su forma estructurada, ya que busca que este sea transformado en matrices y vectores. Para el análisis del suicidio las características son importantes para clasificar el vocabulario general de los tweets relacionados con la ideación suicida recopilados en el paso anterior. Esta extracción de características se consigue con técnicas como la división de palabras o tokenización, eliminando palabras , palabras vacías o lematización, creando un vector de características por cada uno de los tweets con un modelado de N-Grama que se utiliza en minería de texto y PLN, en la creación de nuestro modelo se optó por seguir 3 diferentes estrategias con los datos recolectados, como el análisis de sentimiento, extracción del suicidio y de frivolidad en el contenido.

Para lograr obtener el sentimiento del tweet se utilizó la librería de Python llamada TextBlob agrupándolo en 3 clases : positivo (1), neutral (0) y negativo (-1), recorriendo el “tweet” palabra a palabra (tokenizadas), y buscando el mayor número de palabras positivas para clasificarlo como positivo, o el mayor número de palabras negativas para clasificarlo como negativo, si no era detectada ninguna de las dos clases era automáticamente clasificada como un comentario neutral.

Para obtener el suicidio se utilizaron expresiones regulares [120] [112] para clasificar el suicidio en una forma binaria: El suicidio (-1) y no suicidio (1). Estas expresiones

regulares fueron acorde a la literatura mejorándolas de tal forma que pueda buscar más palabras y/o frases acordes al contenido del tweet obteniendo un resultado favorable, con el uso de la librería *re* y el método *search* se logró identificar las dos clases propuestas para este fin. La expresión regular adaptada y modificada es la siguiente:

```
'.(cutting|depres|sui|these|bad|sa|anx|stre|wake|kill|mental|death|die|li[vf]|worth|tired|
slee|ready|be|never).+(thoughts|feel|plan|pact|up|self|health|note|injury|point|livi|forev|
jum|dea)?.+|.+(can.+|(don.+|do
not)|take|isn.+|slee).+(go|live|anymo|cop|alive|worth|want|wake).+|.+(end.+|(all|it|li[vf]
).+|.+(wan\w.+d[ie]).+|(wan\w.+e[nd]).+|.+(sui|depres|anxie|sa|stre).+'
```

Para detectar comentarios frívolos se utilizaron expresiones regulares [120] [112] para clasificar de forma binaria si existe en el comentario del tweet, un comentario frívolo (-5) o no frívolo (5). Esta expresión regular fue mejorada y adaptada de tal forma que pueda detectar este tipo de contenido, con el uso de la librería *re* y el método *search* se logró identificar las dos clases propuestas para este fin. La expresión regular adaptada y modificada es la siguiente:

```
.(kill|hat|throw).*.+(fuc|f **k).+|.+(you|boy|girl).+(friend)|.+(just).+(like).+
```

Con estas estrategias para clasificar los datos, logramos generar un conjunto de datos con los vectores de palabras por cada tweet preprocesado con las técnicas de PLN, para poder hacer su transformación se utilizó *word2vec* [121], por su simplicidad y eficacia ya que se ocupa del contexto y coherencia de las palabras con lo cual logramos la obtención del conjunto de datos final de vectores numéricos, donde la longitud del vector de variables independientes se generó basándonos en el tamaño inicial de un tweet de 140 palabras, asumiendo que todos los tweet pueden o no tener esa longitud, considerando que de la estrategia de eliminación de ruido detallada en la etapa anterior nos ayudaría a obtener un “tweet” más corto en contenido, para las variables dependientes se usaron las clases del suicidio (1) y (-1) .

5.2.6 Generación del Modelo de Aprendizaje Máquina

Con el conjunto de datos final obtenido con el *word2vec* en la etapa anterior, generamos el modelo de aprendizaje máquina con los clasificadores supervisados Logistic Regression (LR), Support Vector Machine (SVM), K-Nearest Neighbors (K-NN), Random Forest (RF) y Multi-layer Perceptron (MLP) utilizando las técnicas de validación cruzada [122] que son aprovechadas para predecir el rendimiento de los modelos de aprendizaje, ocupando un conjunto de datos para la validación que son independientes a los datos utilizados para el entrenamiento del modelo, entre las técnicas utilizadas se puede encontrar el entrenamiento de retención o Tabla Testing (Training Holdout Cross Validation) [123] que divide el conjunto de datos en entrenamiento (70%) y otro en validación (30%), o como las técnicas exhaustivas de Leave-One-Out el cual utiliza 1 punto de muestra como conjunto de validación y las n-1 muestras restantes se utilizan

como un conjunto de entrenamiento, así como las técnicas no exhaustiva como K-Fold que divide en K partes de igual tamaño, donde un fold se utiliza como un conjunto de validación y los restantes K-1 Folds se utilizaran como un conjunto de entrenamiento, las cuales fueron utilizadas en nuestra investigación para la corroboración de la predicción de nuestro modelo.

5.2.7 Rendimiento en el tiempo del Modelo

La optimización del tiempo de procesamiento en extracción de patrones relacionados con el suicidio fue logrado con la inteligencia de enjambres, buscando mejorar los tiempos de procesamiento, por lo cual es clave tomar en consideración el problema de la programación de tareas considerando un conjunto de procesadores $P = \{p_1, p_2, p_3, \dots, p_m\}$ y un conjunto de tareas (tweets) a procesar $T = \{t_1, t_2, t_3, \dots, t_n\}$ que consiste en asignar una tarea $t_1 \in T$ a un procesador de tal forma que se esté minimizando el tiempo total de ejecución, maximizando la utilización de los procesadores planificando la correcta distribución de las cargas en una arquitectura multicore, para poder resolver este problema de planificación y encontrar un mínimo costo computacional se deben de considerar las siguientes técnicas que serán parte de la función de aptitud utilizada por PSO.

Maxspan

Es la cantidad máxima de tiempo de finalización de procesamiento de las tareas asignadas actualmente y las tareas asignadas a los procesadores del sistema (P_x) y es obtenida por la siguiente ecuación.

$$maxspan = carga\ actual\ P_x + nueva\ carga\ P_x \quad (5.1)$$

Utilización media del procesador

La utilización media del procesador (average processor utilization, *apu*, por sus siglas en inglés) es otro componente necesario para la correcta planificación de las tareas, evitando una elevada carga en los procesadores, esto se puede obtener con la siguiente ecuación.

$$apu = \frac{P_x(tiempo\ de\ ejecución)}{maxspan} \quad (5.2)$$

Cola de procesadores Aceptables

La cola de procesadores aceptables es un factor importante para considerar en la planificación de tareas para maximizar el rendimiento y eficiencia del sistema, se debe considerar un número limitado de colas de procesamiento para minimizar la sobrecarga en la gestión de tareas y maximizar la eficiencia de la ejecución, esto debe realizarse a cada uno de los procesadores para validar si no se están sobrecargados, para lograr esto se utilizan umbrales de carga livianos y pesados [124], esto se puede obtener con la siguiente ecuación

$$cpa = \frac{\# \text{cola de procesadores aceptables}}{\# \text{procesadores}} \quad (5.3)$$

Función de Aptitud

La unión de estos métodos es necesaria para poder lograr la correcta planificación de las tareas en los procesadores y la optimización de tiempo con el PSO, utilizados como componentes principales de la función objetivo, la cual se detalla en la ecuación.

$$fitness = \left(\left(\left(\frac{1}{maxspan} \right) * apu \right) * cpa \right) \quad (5.4)$$

Restricciones

Se deben de considerar las siguientes consideraciones restricciones para la planificación de las tareas:

- Todas las tareas enviadas son independientes
- Una misma tarea no puede ser asignada a más de un procesador

5.2.8 Métricas y métodos de evaluación de resultados

Para determinar el rendimiento de los algoritmos y de su configuración, es necesario disponer de una serie de medidas que permitan evaluar objetivamente su eficacia a la hora de clasificar los ejemplos dados. Para ello, es importante considerar no sólo las muestras clasificadas correcta e incorrectamente, sino también aquellas que, de clasificarse incorrectamente podrían haber sido etiquetadas correctamente.

Métricas de Evaluación

Para poder evaluar este desempeño, aplicaremos los cuatro posibles estados a clasificar, con las métricas de evaluación [125] de la siguiente forma:

- Verdaderos positivos (VP)
 - Son aquellas muestras con clases positivas han sido clasificadas como positivas (clasificados correctamente).
- Verdaderos negativos (VN)
 - Son aquellas muestras con clases negativas que han sido clasificadas como negativas (clasificadas correctamente).
- Falsos positivos (FP)
 - Son aquellas muestras con clases negativas que han sido clasificadas como positivas (clasificadas incorrectamente).
- Falsas negativos (FN)
 - Son aquellas muestras con clases positiva que han sido clasificadas como positiva (clasificadas incorrectamente).

Teniendo en cuenta los estados anteriores, podemos definir las siguientes medidas que serán usadas para evaluar nuestro modelo:

La Exactitud: Es la medida del rendimiento más simple e intuitiva y representa la razón entre las predicciones correctas sobre el total de predicciones realizadas, y se logra con la siguiente fórmula:

$$Exactitud = \frac{VP+VN}{(VP + FP+ VN+FN)} \quad (5.5)$$

La Precisión: [126] Se utiliza para poder medir la CALIDAD que el modelo es capaz de identificar en la tarea de clasificar, y se logra con la siguiente fórmula:

$$Precision = \frac{VP}{(VP + FP)} \quad (5.6)$$

El Recall (exhaustividad o sensibilidad): Se utiliza para poder medir la CANTIDAD que el modelo es capaz de identificar en la tarea de clasificar, y se logra con la siguiente fórmula:

$$Recall = \frac{VP}{(VP + FN)} \quad (5.7)$$

El F1-Score: Se utiliza para poder combinar las medidas de precisión y recall en un solo valor, ya que hace más fácil el poder comparar el RENDIMIENTO combinado entre varias soluciones y se logra con la siguiente fórmula:

$$F1 - Score = 2 \left(\frac{Precision}{(Precision+Recall)} \right) \quad (5.8)$$

Para modelos que cuenten con más de dos clases como es el caso de nuestra investigación, se debe calcular cada una de las métricas anteriores por cada clase de modelo y combinarlas entre ellas para obtener una media global.

Macro promedio: Se utiliza para calcular las medidas de cada clase, y se logra con la siguiente fórmula:

$$macro - promedio = \frac{\sum_{i=1}^n Precision_i}{n} \quad (5.9)$$

$$macro - recall = \frac{\sum_{i=1}^n Recall_i}{n} \quad (5.10)$$

Una desventaja con estas medidas es que no se tiene en consideración la desigualdad que puede existir entre el número de ejemplos de cada clase, por lo que los resultados obtenidos pueden no ser fiables si se presenta este escenario.

Micro promedio: Se utiliza para calcular los indicadores, y se logra con la siguiente fórmula:

$$micro - promedio = \frac{\sum_{i=1}^n VP_i}{\sum_{i=1}^n VP_i + \sum_{i=1}^n FP_i} \quad (5.11)$$

$$micro - recall = \frac{\sum_{i=1}^n VP_i}{\sum_{i=1}^n VP_i + \sum_{i=1}^n FN_i} \quad (5.12)$$

La desventaja en esta técnica es que se tiene en consideración la desigualdad entre los números de ejemplos de cada clase, pero si existen más de dos clases distintas, se tendría que modificar $\sum_{i=1}^1 VP_i = \sum_{i=1}^n FN_i$, por lo que la precisión, el recall y el F1-Score, tomarán siempre los mismos valores.

El promedio ponderado: Se utiliza para resolver el desequilibrio del conjunto de datos con más de dos clases distintas, y se logra con la siguiente fórmula:

$$precision - ponderada = \frac{\sum_{i=1}^n Precision_i * P_i}{n} \quad (5.13)$$

$$recall - ponderada = \frac{\sum_{i=1}^n Recall_i * P_i}{n} \quad (5.14)$$

Matriz de confusión

La evaluación del desempeño en modelo como la literatura recomienda, hace uso de una matriz de confusión [123], que contiene información sobre las clasificaciones reales y estimadas, esta matriz es NxN, donde las filas se nombran según las clases de entrada reales y las columnas son las previstas por el modelo, y son utilizadas para detallar de forma explícita cuando una clase es confundida con otra.

Entrenamiento de modelos de ML

Es una técnica que se utiliza para ajustar los parámetros del modelo para que pueda “aprender” a realizar una trabajo en específico utilizando los datos de nuestra metodología detallada en los puntos anteriores, este aprendizaje se basa en minimizar el error o maximizar la precisión, esto se logra dividiendo el conjunto de datos en dos partes. La primera parte del conjunto de datos es utilizado para el entrenamiento y el resto para las pruebas.

Evaluación de modelos de ML

Es una técnica que se utiliza para evaluar la capacidad de la generalización de un modelo dividiendo los datos en subconjuntos de entrenamiento y prueba, entre las técnicas más utilizadas por la literatura podemos encontrar:

La técnica de entrenamiento de retención o Tabla Testing (Training Holdout Cross Validation) [123]: Es la técnica de validación cruzada más simple. Esta técnica se basa en dividir los datos en un conjunto de entrenamiento y otro de pruebas. Donde el conjunto de entrenamiento se utiliza para construir el modelo de predicción, el cual se valida con el conjunto de datos que se destinó para las pruebas, esto logra que la predicción sea

realizada con la comparación entre los datos predichos y los datos reales, ver Figura 5.2. Esta técnica tiene una elevada variación, esto por los resultados que pueden salir dependiendo de la división inicial del conjunto de datos.

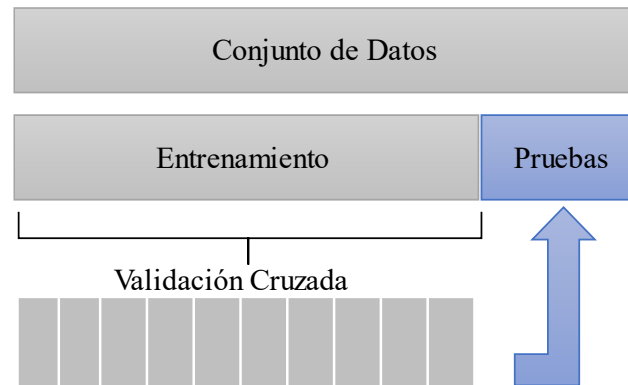


Figura 5.2 Técnica Tabla Testing ó Training Holdout

La técnica de validación cruzada (Cross Validation en inglés): Es un método de evaluación en modelos en los que se trata un gran número de datos para establecer la independencia de los resultados (la predicción) de la partición inicial del conjunto de datos en el entrenamiento (training en inglés) , pruebas (test en inglés). Es un método utilizado en Machine Learning, para garantizar la independencia de los resultados de la partición inicial en los datos de entrada.

La técnica de validación cruzada K -Fold: Dividen los datos en K subconjuntos y se repite la técnica de tabla testing K veces de manera que en cada iteración, se escoge uno de los K subconjuntos como el conjunto de datos para las pruebas y los $K-1$ restantes, se utilizan como conjunto de datos de entrenamiento. La ventaja de esta técnica es que el resultado depende menos de la división que se hace de los datos en los conjuntos de datos de entrenamiento y prueba. Una vez realizados todos los modelos, se obtendrá el resultado final como la media aritmética de las K iteraciones ver Figura 5.3.

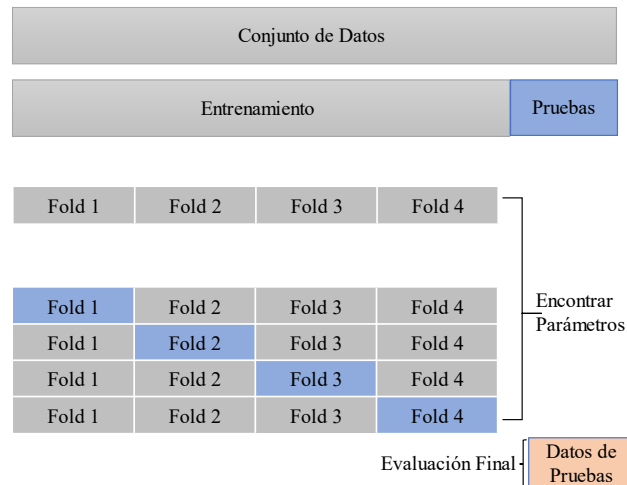


Figura 5.3 Técnica K-Fold

La técnica de validación cruzada Leave One Out: Utiliza 1 punto de muestra como conjunto de validación y las $n-1$ muestras restantes se utilizan como un conjunto de entrenamiento ver Figura 5.4, la principal desventaja de esta técnica es el alto costo computacional que se tiene, ya que el reajuste y la validación en tantas interacciones es muy complicado.



Figura 5.4 Técnica Leave One Out

5.3 Materiales

Sistema Operativo

Windows 10 desarrollado por Microsoft fue el sistema operativo utilizado en esta investigación, donde se realizó el desarrollo y pruebas del sistema, respaldando todo el código en los repositorios GitHub y en OneDrive la documentación, utilizando un equipo de cómputo ZBOOK con procesador Intel® Xeon® E-2176M CPU @ 2.70GHz 2.71GHz , memoria de 32GB y disco duro de 1TB de estado sólido.

Python (v3.8.x)

Es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. Se trata de un lenguaje de programación multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma.

Scikit-learn (v0.24.2)

Biblioteca de Python que soporta el aprendizaje supervisado y no supervisado. También proporciona varias herramientas para el ajuste de modelos, el preprocesamiento de datos, la selección y evaluación de modelos y muchas otras utilidades.

Pandas (v1.3.1)

Biblioteca de Python pretende ser el bloque de construcción de alto nivel fundamental para realizar análisis de datos prácticos y reales. Además, tiene el objetivo más amplio de convertirse en la herramienta de análisis/manipulación de datos de código abierto más potente y flexible disponible en cualquier lenguaje.

TextBlob (v0.16.0)

Biblioteca de Python (2 y 3) para el procesamiento de datos textuales. Ofrece una API sencilla para sumergirse en tareas comunes de PNL, como el etiquetado de partes del habla, la extracción de frases sustantivas, el análisis de sentimientos, la clasificación y la traducción.

Transferencia de Estado Representacional (REST)

Es un nuevo enfoque de desarrollo de proyectos y servicios web fue definido por Fielding, R. T. [1] , el padre de la especificación del Protocolo de transferencia de hipertexto (HTTP) y uno de los referentes internacionales en todo lo relacionado con la Arquitectura de Redes.

REST API de Twitter

Ofrece a los desarrolladores el acceso a los datos de Twitter. Todas las operaciones que se pueden hacer vía web son posibles realizarlas desde el API. Dependiendo de la operación requiere o no autenticación, con el mismo criterio que en el acceso web. Soporta los formatos: XML, JSON, RSS, ATOM.

5.4 Conclusiones del capítulo

En este capítulo se detallaron los pasos de la metodología diseñada para optimizar el tiempo de procesamiento en la detección de patrones relacionados con el suicidio en redes sociales ha demostrado ser eficaz para acelerar el preprocesamiento de identificación de posibles amenazas y señales de advertencia en las redes sociales. Al combinar procesamiento de lenguaje natural, expresiones regulares, análisis de sentimientos y algoritmos de aprendizaje automático, hemos podido reducir significativamente el tiempo que lleva detectar mensajes de alarma. Esta optimización es fundamental para brindar respuestas más rápidas y oportunas para prevenir posibles suicidios y apoyar a las personas en esta situación. Esta metodología basa su enfoque en optimizar el tiempo en la identificación de patrones asociados al suicidio en Twitter, ofreciendo una visión más profunda y contextualizada de la conversación en línea que puede ser de gran utilidad en diversas aplicaciones y áreas de investigación.

Capítulo 6. RESULTADOS Y DISCUSIÓN

En este capítulo se discutirán los resultados obtenidos de la comparación realizada entre las diferentes técnicas de extracción de patrones y la precisión de los algoritmos seleccionados para validar la precisión de nuestro modelo. Haciendo una comparación de los resultados obtenidos con los tiempos entre el preprocesamiento de datos de una forma secuencial y con el uso del algoritmo bioinspirado seleccionado para esta investigación.

6.1 Planificación de tareas

La planificación de las tareas es el proceso de equilibrar la asignación de procesos o tareas en los procesadores o en nodos en un sistema distribuido. Es fundamental esta planificación para poder optimizar los recursos del sistema para garantizar el mejoramiento del rendimiento, haciendo que esta planificación pueda hacer que las tareas se ejecuten de una forma más eficiente, minimizando los tiempos de espera.

6.2 Técnica de Ventana deslizante

Una de las técnicas utilizadas para realizar la planificación de las tareas es la ventana deslizante (Sliding Window en inglés). Esta técnica es utilizada para realizar la gestión y poder optimizar la planificación de las tareas o procesos en ejecución del sistema. La ventana deslizante segmenta el número de tareas de una forma aleatoria del conjunto de datos de forma dinámica, generando listas aleatorias de tareas (tweets), como se puede observar en la Figura 6.1, a medida de que el procesamiento del sistema avanza, la ventana ejecuta un “deslizamiento” para poder entregar la siguiente secuencia de tareas que se pondrán a disposición del sistema considerando a todos los procesadores con la misma prioridad de asignación.

No.	Venita desizante																																																
1	9	48	37	20	31	4	10	15	8	23	35	14	13	26	6	24	49	29	45	16	39	18	11	34	25	41	5	21	47	46	12	7	36	33	19	27	22	38	40	28	2	30	3	44	17	42	32		
2	22	18	46	29	44	19	10	2	22	40	43	12	4	35	41	46	37	16	2	27	41	26	4	41	32	22	9	29	33	7	34	22	37	45	13	8	3	19	2	46	42	9	5	42	35	9			
3	20	2	36	13	5	32	46	8	20	27	4	24	8	37	36	27	16	21	17	36	36	38	11	46	7	42	21	24	41	23	18	37	20	27	45	33	6	38	32	18	27	17	32	9	15	14	33		
4	38	27	40	17	25	46	29	14	29	2	36	44	43	27	25	21	24	44	36	8	5	18	42	7	45	27	39	31	9	33	22	3	40	22	34	21	13	44	25	9	30	35	36	34	18	40	24	30	
5	32	5	19	18	4	16	32	37	23	39	7	34	23	16	30	2	26	14	38	26	6	28	31	15	21	28	16	28	15	24	22	13	40	25	27	45	42	41	21	29	45	28	8	23	25	42	24	26	
6	17	12	32	42	11	30	34	44	40	35	38	40	30	8	5	41	8	35	2	4	32	32	40	20	29	6	35	42	24	13	22	40	17	21	29	43	26	19	21	49	35	26	34	32	34	24	4		
7	22	18	46	29	44	19	10	2	22	40	43	12	4	35	41	46	37	16	2	27	41	26	4	41	32	22	9	29	33	7	34	22	37	45	13	8	6	8	19	2	46	42	9	5	42	35	9		
8	25	8	3	27	29	34	4	39	8	45	21	27	15	28	9	39	40	13	42	25	22	35	43	42	15	23	16	15	25	28	18	22	24	6	43	9	42	7	28	26	13	19	38	9	12	9	40	11	
9	6	37	41	20	30	33	33	30	33	39	34	5	13	8	17	5	18	6	31	10	46	6	2	43	6	15	19	12	33	34	16	12	38	13	10	3	27	5	39	25	12	42	26	27	17				
10	41	31	16	29	7	7	4	23	34	24	42	37	28	45	2	5	16	24	24	15	5	33	38	8	2	14	30	26	3	22	35	7	19	37	44	19	16	8	36	30	11	20	22	15	42	8	21		
11	14	15	4	20	18	19	5	28	42	9	25	15	16	5	30	45	41	25	38	38	15	2	45	27	12	32	28	8	30	6	42	39	28	3	29	26	6	6	7	17	5	6	31	12	3	34	4		
12	24	16	35	9	3	20	39	3	17	37	32	14	38	3	15	5	27	127	36	17	4	25	22	22	3	17	7	46	28	28	2	31	6	42	39	28	2	9	31	6	18	28	13	38	38	37	28	37	27
13	32	20	31	4	40	25	12	39	22	38	4	15	36	23	38	44	24	42	34	33	6	28	41	7	27	44	9	39	38	35	28	26	24	34	36	22	4	2	8	31	10	42	17	19	45	41	37		
14	31	20	37	2	45	7	35	30	39	28	23	14	9	37	12	15	19	30	36	14	5	45	15	9	30	26	40	26	30	12	44	39	3	15	19	11	43	4	24	20	42	35	7	8	9	23	24		
15	26	29	35	42	8	19	28	21	29	37	34	19	14	3	39	16	21	44	44	25	13	42	31	32	40	20	19	32	10	41	43	23	18	37	24	18	14	45	10	3	46	45	30	35	4	24	8		
16	6	9	15	41	23	2	19	43	42	21	7	25	42	14	21	24	40	38	40	24	5	41	14	25	41	15	46	42	2	44	12	29	40	38	36	20	8	30	29	11	40	34	37	14	36	43	44		
17	34	13	4	29	8	7	37	45	34	43	39	42	13	23	14	25	27	36	18	43	18	46	41	43	44	26	4	33	37	46	14	4	24	3	32	44	45	44	8	40	6	43	43	17	3	26	21	5	
18	7	4	40	20	30	34	26	41	46	44	2	2	26	10	11	11	44	44	23	23	44	30	24	9	49	26	31	21	44	47	4	9	24	21	20	33	13	43	40	42	7	40	42	40	41	6	26	13	
85	34	22	30	19	40	27	33	16	17	33	29	29	22	39	36	38	14	40	46	29	8	34	46	29	28	9	32	24	28	13	32	3	36	14	46	14	10	33	43	27	13	38	40	17	37	32	34	29	

Figura 6.1 Ventana Deslizante.

6.3 Experimento de Planificación de Tareas con AG

El proceso de planificación de tareas utilizando el algoritmo genético comprende varias técnicas. Estas técnicas demuestran cómo este enfoque puede optimizar los recursos del sistema y mejorar su rendimiento al disminuir los tiempos de espera. A partir de la primera "ventana" de tareas generada por la técnica de ventana deslizante, se obtiene una lista de tareas que se trasladará a través de los diferentes pasos y técnicas del algoritmo genético.

Técnica de permutación de tareas: Se refiere a la técnica de reordenar tareas entregadas de la primera “ventana” de la técnica de la ventana deslizante, de esta manera buscar la optimización del sistema, generando 24 cromosomas a partir de esta ventana, como se puede observar en la Figura 6.2.

Permutación de Tareas																																																
	9	48	37	20	31	4	10	15	8	23	35	14	13	26	6	24	49	29	45	16	39	18	11	43	34	25	41	5	21	47	46	12	7	36	33	19	27	22	38	40	28	2	30	3	44	17	42	32
Cromosomas	10	16	33	18	28	14	42	41	49	24	25	22	45	31	44	35	11	8	39	7	40	17	3	2	30	13	29	36	34	4	32	47	12	6	46	48	23	20	27	26	21	9	38	43	37	19	5	15
	13	11	12	10	16	43	26	7	20	5	33	36	38	22	32	46	44	19	35	2	39	40	48	6	25	27	23	24	14	8	21	30	34	4	31	37	45	29	47	18	42	3	49	28	17	41	9	15
	13	7	27	40	37	5	44	30	12	2	14	36	41	10	46	4	9	42	16	8	23	20	15	48	47	11	43	35	3	38	28	49	45	17	19	33	34	6	18	24	31	29	25	21	22	26	39	32
	14	49	17	33	40	34	12	22	7	28	18	43	19	41	47	11	44	8	25	4	45	30	48	5	13	32	37	20	23	35	24	36	9	46	39	29	2	21	10	15	3	27	31	38	16	6	42	26
	37	49	10	3	38	40	11	20	5	42	36	23	15	14	18	46	30	21	4	26	16	27	33	22	28	13	31	29	34	43	39	17	24	45	2	9	44	47	41	25	32	7	48	6	35	8	12	19
	47	25	32	9	20	43	29	45	7	16	26	18	33	8	3	17	46	12	41	4	5	10	44	30	37	38	36	23	2	40	24	19	34	27	28	13	21	42	49	48	22	35	39	14	31	6	15	11
	21	34	45	31	42	4	12	3	39	46	19	41	10	11	40	38	35	8	15	6	28	47	26	16	44	14	13	29	20	18	36	27	24	43	5	37	7	2	22	25	9	33	23	30	48	49	32	17
	9	41	20	21	6	27	26	4	15	39	47	23	38	48	7	3	32	11	22	45	18	25	14	34	28	8	36	46	31	37	44	43	10	19	35	40	24	33	30	12	5	42	13	17	29	2	49	16
	7	35	15	44	14	9	16	41	27	20	13	4	19	32	2	48	49	33	8	37	46	31	24	22	12	25	6	3	28	45	42	40	21	26	29	23	38	43	10	30	34	36	39	11	17	47	18	5
	47	6	41	44	42	12	20	48	14	2	23	19	18	11	10	17	25	5	13	39	40	34	37	45	46	7	27	36	9	49	38	32	33	24	16	3	43	30	28	35	4	26	29	8	21	15	31	22
	11	32	16	26	15	44	49	17	38	2	19	7	14	45	5	3	43	42	33	4	31	23	9	21	6	25	13	48	20	37	10	40	34	27	24	22	18	35	46	36	39	29	41	47	28	8	30	12
	48	27	35	4	29	32	38	15	3	49	41	34	28	24	46	6	31	33	22	11	19	26	30	39	12	23	16	20	43	17	8	36	13	2	47	44	45	9	37	18	40	5	25	7	21	42	14	10
	29	41	32	8	2	10	23	49	39	24	31	14	45	11	27	47	15	26	36	28	16	46	42	22	34	44	13	19	33	48	38	43	35	9	21	5	6	37	20	4	3	30	12	40	18	25	17	7
	8	14	10	48	41	45	43	5	28	29	22	39	20	37	35	33	16	17	30	31	15	13	32	6	9	25	42	2	3	24	38	27	34	19	11	21	12	49	44	18	7	26	40	4	47	23	46	36
	41	28	34	18	30	29	44	33	39	11	42	45	5	3	49	31	26	25	16	37	46	47	48	8	43	2	4	15	24	19	40	38	35	10	23	27	36	20	9	6	17	13	7	14	32	12	22	21
	31	19	7	27	42	24	43	8	13	20	28	35	46	36	15	14	3	22	23	25	37	26	18	32	29	47	17	10	44	49	12	16	41	5	9	48	45	6	33	2	4	30	39	40	34	11	38	21
	34	18	44	16	46	23	5	27	49	48	10	36	8	19	33	47	30	9	26	29	17	21	20	28	12	43	25	42	37	39	45	24	15	3	4	40	14	32	41	22	7	13	38	31	2	6	11	35
	22	18	6	23	14	45	39	38	46	7	26	11	15	48	29	20	24	49	27	36	19	16	47	17	25	4	3	30	41	44	5	2	21	10	28	40	8	31	42	34	33	13	9	37	43	32	35	12
	30	24	9	4	15	42	38	6	27	41	31	26	44	8	35	43	17	25	2	10	47	12	40	46	23	19	5	7	48	36	18	33	45	49	14	11	20	32	22	21	28	29	39	16	13	34	37	3
	49	4	9	18	48	36	6	39	30	8	31	28	25	43	14	26	15	44	38	27	42	5	2	3	24	22	41	11	40	32	45	37	7	29	20	16	21	34	35	13	33	10	23	47	19	17	12	46
	15	17	10	28	12	37	47	9	26	19	25	2	7	18	39	34	42	41	3	43	36	45	31	24	13	49	21	46	6	33	4	11	20	32	38	16	29	23	44	5	27	35	48	30	22	8	14	40
	12	24	27	47	34	9	41	32	29	37	11	6	38	35	48	4	15	42	33	25	3	2	39	8	18	19	14	43	46	28	10	16	49	13	23	30	22	5	45	20	17	7	31	21	40	44	26	36
	20	44	29	6	2	41	42	43	48	37	31	38	22	5	39	28	30	3	16	8	46	11	25	24	19	32	34	33	40	26	23	4	21	13	49	36	47	10	7	12	17	18	45	9	15	35	14	27

Figura 6.2 Permutación de cromosomas.

Estado Actual de los Procesadores

En este paso, se recolecta el estado actual de los procesadores, para visualizar qué tareas se encuentran ejecutando en el sistema por cada uno de los procesadores, como se visualiza en la Tabla 6.1.

Tabla 6.1 Estado del Actual de los Procesadores.

Núm. Procesador	Tareas Actuales	Cola del Procesador	Longitud total de la Cola
1	2(7)	4(8), 3(7), 6(9), 5(6), 8(10)	47
2	10(13)	11(12), 13(9), 9(7), 15(12)	53
3	14(11)	12(7), 17(13), 19(6)	37

4	16(8)	18(9), 21(8), 23(11), 25(13)	49
5	20(8)	0	8
6	22(15)	24(7), 26(9), 28(13), 30(11)	55
7	31(6)	33(8), 35(11), 37(7)	32
8	39(4)	32(12), 34(16)	32
9	36(12)	38(5), 40(12), 42(6), 44(9)	44
10	46(20)	48(15), 41(13), 43(8), 45(9)	65
11	47(9)	49(8), 51(6)	23
12	52(14)	53(15)	29

Nuevas tareas que llegan para ser planificadas

En este paso se detalla la llegada del nuevo conjunto de tareas (tweets) [9, 48, 37, 20, 31, 4, 10, 15, 8, 23, 35, 14, 13, 26, 6, 24, 49, 29, 45, 16, 39, 18, 11, 43, 34, 25, 41, 5, 21, 47, 46, 12, 7, 36, 33, 19, 27, 22, 38, 40, 28, 2, 30, 3, 44, 17, 42, 32] después de haber sido permutadas en con la técnica de permutación de tareas.

Asignación de las tareas por procesador.

Basándose en el conjunto de nuevas tareas que arribaron al sistema, se ejecuta la asignación de tareas, acorde al arribo de cada una de ellas, ejemplo, la tarea 10, es asignada en el procesador 1, la tarea 8 es asignada al procesador 2 y así hasta terminar con el procesador 12, retornando al procesador 1, asignándole la tarea 12, hasta posicionar todas las nuevas tareas en los procesadores como se puede observar en la Tabla 6.2. En la columna “Nueva Carga” se establecen las nuevas tareas que serán agregadas a los procesadores.

Tabla 6.2 Tareas asignadas en los procesadores

Núm. Procesador	Nuevas tareas asignadas	Nueva Carga
1	70(9),82(13),94(34),66(27)	83
2	71(48),83(26),95(25),67(22)	121
3	72(37),84(6),96(41),68(48)	122
4	73(20),85(24),97(5),69(40)	89
5	74(31),86(49),98(21),54(28)	129
6	75(4),87(29),99(47),55(2)	82
7	76(10),88(45),60(46),56(30)	131
8	77(15),89(16),61(12),57(3)	46
9	78(8),90(39),62(7),58(44)	98
10	79(23),91(18),63(36),59(17)	94
11	80(35),92(11),64(33),23(43)	121
12	81(14),93(43),65(19),25(32)	108

Determinar la carga promedio de los procesadores

El siguiente para es determinar los nuevos umbrales basados en la carga actual del sistema y las nuevas tareas que arribaron y se planificaron para que se programe a integrarse al sistema, para obtenerlo, utilizaremos la siguiente fórmula:

$$lave = ((\sum cl + \sum nl) / n) \quad (6.1)$$

Donde, se obtendrá el porcentaje de la carga (*lave*), realizando la sumatoria de la carga actual (*cl*) y la nueva carga (*nl*) dividida entre el número de procesadores (*n*).

$$cl = 47 + 53 + 37 + 49 + 8 + 55 + 32 + 32 + 44 + 65 + 23 + 29$$

$$nl = 83 + 121 + 122 + 89 + 129 + 82 + 131 + 46 + 98 + 94 + 121 + 108$$

$$n = 12$$

Sustituyendo:

$$lave = (474 + 1224) / 12.$$

$$lave = 141.5$$

Umbrales Pesados y Livianos

Con el resultado del porcentaje de la carga (*lave*), podemos calcular los umbrales con las siguientes ecuaciones:

$$up = p * lave \quad (6.2)$$

$$ul = l * lave \quad (6.3)$$

Donde, *up* es el umbral pesado, *ul* es el umbral liviano, *p* es la constante pesada, *l* la constante liviana.

Sustituyendo:

$$up = (1.2 * 141.5)$$

$$up = 170 \text{ (Redondeado)}$$

$$ul = (0.8 * 141.5)$$

$$ul = 113 \text{ (Redondeado)}$$

Con estos resultados, se determina que el nuevo umbral pesado será de 170 y el umbral liviano quedará en 113.

Encontrar Carga Aceptable

En este paso se seleccionarán los procesadores aceptables para recibir carga, lo cual se puede calcular restando las colas de los procesadores actuales, utilizando el nuevo umbral pesado, logrando esto con la siguiente ecuación:

$$pa = (nl_{px} \leq up - cl_{px}) \quad (6.4)$$

Donde:

nl_{px} = Carga aceptable del procesador 1

up = umbral pesado

cl_{px} = Longitud total de la Cola del procesador 1

Sustituyendo:

$$83 \leq 170 - 47$$

$$83 \leq 123$$

De manera similar, se tiene que realizar la validación por cada uno de los procesadores, dando como resultado que el procesador 2 no pueden recibir carga, siendo los

2	0.041470588	0.082139763	0.185426504
3	0.040428062	0.080074857	0.265501361
4	0.039605893	0.078446409	0.34394777
5	0.037656738	0.074585766	0.418533536
6	0.029844553	0.059112364	0.4776459
7	0.027283951	0.054040643	0.531686543
8	0.025060701	0.049637107	0.58132365
9	0.023308015	0.046165606	0.627489256
10	0.022563656	0.044691273	0.672180529
11	0.021158463	0.041908043	0.714088571
12	0.016070866	0.031831165	0.745919736
13	0.0148324	0.029378166	0.775297902
14	0.013280886	0.026305121	0.801603023
15	0.013162893	0.026071414	0.827674437
16	0.012469008	0.024697055	0.852371492
17	0.011805257	0.02338238	0.875753872
18	0.011667477	0.023109482	0.898863353
19	0.011292329	0.022366435	0.921229788
20	0.010064	0.019933515	0.941163302
21	0.00972018	0.019252519	0.960415822
22	0.008038521	0.015921699	0.97633752
23	0.006435719	0.012747068	0.989084588
24	0.005510955	0.010915412	1
	0.50487835	1	

El resultado obtenido con el giro de la ruleta, se puede observar en la Figura 6.4 que el cromosoma 1 y 15 fueron los seleccionados para las siguientes técnicas del algoritmo genético.

Cromosoma	Tareas																																															
X	9	48	37	20	31	4	10	15	8	23	35	14	13	26	6	24	49	29	45	16	39	18	11	43	34	25	41	5	21	47	46	12	7	36	33	19	27	22	38	40	28	2	30	3	44	17	42	32
Y	8	14	10	48	41	45	43	5	28	29	22	39	20	37	35	33	16	17	30	31	15	13	32	6	9	25	42	2	3	24	38	27	34	19	11	21	12	49	44	18	7	26	40	4	47	23	46	36

Figura 6.4 Selección de dos cromosomas con tecnica de ruleta.

Cruzamiento

Utilizando la técnica de cruzamiento en ciclos detallado en el Capítulo 4, en la sección 4.2.1, se logró el siguiente resultado reflejado en la Figura 6.5. donde se decidió iniciar con el padre X y luego con el padre Y, logrando obtener la descendencia, que será parte de la nueva población.

Solicitando y Actualizando la ventana deslizando

Cuando todo el proceso anterior realiza, se hace la solicitud a la ventana deslizando que asigne nuevas tareas, sumando las tareas que no fueron procesadas por el procesador 2, por no poder aceptar carga, evitando tener pérdida de tareas por nuestro sistema. Con este nuevo conjunto de tareas para ser planificada, se tiene que regresar a los pasos anteriores y repetirlos nuevamente hasta terminar todo el conjunto de datos recolectado, como se observa en la Tabla 6.4.

Tabla 6.4 Nuevo contenido de Ventana Deslizando

Tareas sin asignar	Nuevas tareas	Tareas para planificar
71(48),83(26),95(25),67(22)	22,18,46,29,44,19,10,2,22,40,43,12,4,35,41,46,37,16,2,27,41,26,4,41,32,22,7,9,29,33,7,34,22,37,45,13,8,6,8,19,2,46,42,9,5,42,35,9	48,26,25,22,22,18,46,29,44,19,10,2,22,40,43,12,4,35,41,46,37,16,2,27,41,26,4,41,32,22,7,9,29,33,7,34,22,37,45,13,8,6,8,19,2,46,42,9,5,42,35,9

De acuerdo con los resultados en esta simulación, se logra observar que el uso de la calibración de carga junto con los umbrales adecuados y el tamaño correcto de iteraciones y posibles soluciones se obtienen resultados cercanos al óptimo, aunque no podemos comparar directamente nuestros resultados en tiempo de preprocesamiento, podemos hacerlo en términos de estrategia de planificación de tareas como la de Zomaya et al. [124], donde podemos observar que el uso correcto de generaciones, poblaciones y una programación de carga flexible del 20% equivalente en los umbrales $H=1.2$ y $L=0.8$, puede lograr aumentar significativamente la utilización de los procesadores. Aunque el uso de la metaheurísticas es completamente diferente se puede estar de acuerdo, que la correcta elección del tamaño de población sigue siendo una tarea de suma importante para este tipo de problemas de planificación de tareas.

6.4 Experimento de Planificación de Tareas con PSO

El proceso de planificación de tareas utilizando PSO incluye diversas técnicas que se detallan a continuación. Estas técnicas demuestran cómo este enfoque optimiza los recursos del sistema y mejora su rendimiento al reducir los tiempos de espera. A partir de la primera lista de tareas generada mediante la técnica de ventana deslizando, se crea una "ventana" de tareas que se transferirá y procesará a lo largo de las diferentes etapas del algoritmo PSO.

Técnica de permutación de tareas: Se refiere a la técnica de reordenar tareas entregadas de la primera “ventana” de la técnica de la ventana deslizante, de esta manera buscar la optimización del sistema, generando 24 partículas a partir de esta ventana, como se puede observar en la Figura 6.7.

Permutación PSO																																																
Partículas	17	17	32	42	11	30	34	44	40	35	38	40	30	8	3	5	41	8	35	2	4	32	32	40	20	29	6	35	42	24	13	22	40	17	6	21	29	43	17	46	35	26	34	32	34	24	34	4
	42	34	41	35	40	38	3	26	29	32	21	35	17	6	8	46	44	34	17	5	8	6	4	11	2	35	35	32	32	34	17	40	20	4	43	32	24	34	40	22	24	13	40	29	42	17	30	30
	32	17	35	4	34	24	6	32	17	40	8	17	43	5	4	34	20	34	22	2	41	8	30	30	3	21	13	24	46	29	29	35	38	44	6	42	35	11	40	40	32	17	35	42	40	34	26	32
	29	41	35	35	40	13	20	26	30	6	4	8	43	17	24	34	30	29	32	32	17	42	4	32	3	6	32	40	34	21	34	40	35	35	17	40	46	2	24	38	42	11	44	8	34	5	22	17
	38	29	29	2	4	40	35	32	40	17	20	42	5	44	42	34	32	8	4	17	17	35	35	43	32	8	32	11	35	22	24	40	24	21	6	30	34	34	46	30	40	13	26	17	41	3	6	34
	32	34	35	17	8	17	34	3	22	34	35	29	44	2	4	32	6	42	24	38	8	26	13	4	30	21	40	40	17	35	41	42	35	40	30	43	46	34	17	32	6	32	29	40	20	11	5	24
	2	3	5	4	8	29	40	32	21	17	42	40	42	4	35	35	17	34	34	29	17	24	40	34	32	43	30	38	8	6	32	41	13	26	35	32	35	46	34	22	44	6	40	11	30	20	24	17
	32	20	6	35	13	17	4	40	17	29	34	17	26	40	22	42	17	29	3	34	8	6	30	35	40	44	42	2	24	34	8	30	43	38	32	32	35	24	41	4	46	21	35	32	40	11	5	34
	35	17	29	26	22	8	11	42	35	34	40	34	30	40	17	32	40	29	43	32	8	35	17	17	5	24	20	13	2	21	34	32	4	4	38	30	46	6	32	41	40	24	35	42	44	6	34	3
	5	32	35	4	38	41	35	17	40	21	42	6	34	34	44	40	34	32	32	46	32	6	8	40	2	8	3	42	17	17	22	30	24	26	13	29	20	30	35	34	35	4	24	29	11	43	17	
	6	20	21	30	35	42	34	35	40	17	2	8	32	46	11	26	13	22	8	38	24	41	32	40	35	40	17	34	34	29	42	32	35	4	4	43	3	24	44	34	17	30	40	29	17	6	5	32
	13	44	34	34	40	26	24	6	42	32	34	30	8	17	6	40	34	4	29	20	21	24	32	17	40	35	41	35	32	38	17	4	5	8	46	35	11	42	43	22	2	32	29	30	35	40	17	3
	21	38	41	5	26	17	34	32	46	32	4	22	24	40	29	42	44	4	24	30	17	11	32	6	3	40	42	35	34	40	17	43	13	2	8	30	29	40	20	34	35	32	35	34	6	17	8	
	40	40	42	26	29	2	4	5	42	17	34	20	17	34	35	22	34	35	24	6	24	34	30	32	38	8	17	44	41	21	29	35	8	46	3	17	11	4	32	30	32	43	6	40	40	13	35	32
	17	44	4	34	34	30	4	24	22	35	40	32	30	32	8	46	35	26	29	5	11	2	21	35	40	34	42	20	34	41	42	38	17	3	32	40	32	24	6	17	13	8	35	43	6	40	29	17
	22	40	32	20	41	40	35	40	11	42	34	34	6	17	17	34	40	30	32	24	29	17	3	17	6	34	38	8	32	29	5	46	26	30	42	21	35	2	4	8	43	35	4	24	32	35	13	44
	35	20	43	35	6	34	4	22	17	32	32	44	24	40	35	17	40	17	32	38	29	40	2	30	11	3	41	42	46	42	4	29	34	26	32	35	24	8	13	8	5	21	34	17	40	30	6	34
	40	35	40	17	46	11	17	34	4	5	34	29	40	34	42	3	40	24	6	8	32	35	26	43	44	29	2	38	34	21	32	17	35	32	6	20	30	4	8	41	35	13	22	24	32	17	30	42
	21	34	17	46	32	40	29	6	32	35	24	6	35	17	20	8	22	42	35	40	34	29	30	13	4	35	17	17	41	40	32	40	11	3	44	26	24	5	42	30	43	34	8	4	32	38	34	2
	35	11	6	6	13	41	20	24	4	32	32	34	3	32	30	26	21	22	40	29	30	38	42	2	43	40	46	35	8	34	29	17	24	4	5	42	17	8	40	40	17	34	32	44	35	34	35	17
	46	29	34	6	32	40	34	22	13	41	30	40	44	8	5	24	35	17	3	35	17	34	43	38	42	26	40	6	42	8	4	34	17	30	32	35	35	4	17	29	11	2	32	32	24	40	20	21
	29	34	38	5	34	44	11	3	34	35	43	40	4	6	29	42	24	17	40	17	30	35	8	42	46	24	40	34	17	13	32	26	40	6	4	35	35	41	32	32	30	21	20	8	17	22	32	2
	21	8	26	13	42	40	40	32	17	24	32	35	43	34	32	24	17	44	5	6	40	32	3	11	30	17	34	8	20	29	35	38	35	6	2	34	41	4	17	22	29	35	4	46	34	40	30	42
	46	24	29	30	6	35	42	13	34	32	40	35	8	29	32	38	40	40	4	30	5	34	35	26	41	8	40	34	21	32	42	17	44	4	17	35	22	32	6	17	24	3	34	17	2	20	43	11

Figura 6.7 Permutación de partículas.

Estado Actual de los Procesadores

En este paso, se recolecta el estado actual de los procesadores, para visualizar que tareas se encuentran ejecutando en el sistema por cada uno de los procesadores, como se visualiza en la Tabla 6.5.

Tabla 6.5 Estado del Actual de los Procesadores.

Núm. Procesador	Tareas Actuales	Cola del Procesador	Longitud total de la Cola
1	2(7)	4(18),3(10),6(19),5(6),8(10)	70
2	10(13)	11(12), 13(26), 9(28), 15(12)	91
3	14(11)	12(26), 17(23), 19(26)	86
4	16(8)	18(15), 2(24), 23(26), 25(33)	96
5	20(8)	2(30), 4(16),12(17)	71
6	22(15)	24(22), 26(19), 28(13), 3(21)	90
7	31(6)	3(15), 35(30), 37(24)	75
8	39(4)	32(12), 34(21), 18(13)	50
9	36(12)	8(15), 40(12), 42(20), 44(13)	72
10	46(20)	48(15), 41(12), 43(18), 5(19)	84
11	47(9)	49(18), 51(21), 9(20),(24)	92
12	52(14)	53(15), 15(16), 8(24),9(25)	94

Nuevas tareas que llegan para ser planificadas

En este paso se detalla la llegada del nuevo conjunto de tareas (tweets) [17,17,32,42,11,30,34,44,40,35,38,40,30,8,3,5,41,8,35,2,4,32,32,40,20,29,6,35,42,24,13,22,40,17,6,21,29,43,17,46,35,26,34,32,34,24,34,4] despues de haber sido permutadas en con la técnica de permutación de tareas.

Asignación de las tareas por procesador.

Basándose en el conjunto de nuevas tareas que arribaron al sistema, se ejecuta la asignación de tareas, acorde al arribo de cada una de ellas, ejemplo, la tarea 10, es asignada en el procesador 1, la tarea 8 es asignada al procesador 2 y así hasta terminar con el procesador 12, retornando al procesador 1, asignándole la tarea 12, hasta posicionar todas las nuevas tareas en los procesadores como se puede observar en la Tabla 6.6. En la columna “Nueva Carga” se establecen las nuevas tareas que serán agregadas a los procesadores.

Tabla 6.6 Tareas asignadas en los procesadores.

Núm. Procesador	Nuevas tareas asignadas	Nueva Carga
1	70(17),82(30),94(20),66(29)	96
2	71(17),83(8),95(29),67(43)	97
3	72(32),84(3),96(6),68(17)	58
4	73(42),85(5),97(35),69(46)	128
5	74(11),86(41),98(42),54(35)	129
6	75(30),87(8),99(24),55(26)	88
7	76(34),88(35),60(13),56(34)	116
8	77(44),89(2),61(22),57(32)	100
9	78(40),90(4),62(40),58(34)	118
10	79(35),91(32),63(17),59(24)	108
11	80(38),92(32),64(6),23(34)	110
12	81(40),93(40),65(21),25(4)	105

Determinar la carga promedio de los procesadores

El siguiente para es determinar los nuevos umbrales basados en la carga actual del sistema y las nuevas tareas que arribaron y se planificaron para que se programe a integrarse al sistema,para obtenerlo, utilizaremos la siguiente fórmula:

$$lave = ((\sum cl + \sum nl) / n) \quad (6.1)$$

Donde, se obtendrá el porcentaje de la carga (*lave*), realizando la sumatoria de la carga actual (*cl*) y la nueva carga (*nl*) dividida entre el número de procesadores (*n*).

$$cl = 70 + 91 + 86 + 96 + 71 + 90 + 75 + 50 + 72 + 84 + 92 + 94$$

$$nl = 96 + 97 + 58 + 128 + 129 + 88 + 116 + 100 + 118 + 108 + 110 + 105$$

$$n = 12$$

Sustituyendo:

$$lave = (971 + 1253) / 12.$$

$$lave = 222.4$$

Umbrales Pesados y Livianos

Con el resultado del porcentaje de la carga (*lave*), podemos calcular los umbrales con las siguientes ecuaciones:

$$up = p * lave \quad (6.2)$$

$$ul = l * lave \quad (6.3)$$

Donde, *up* es el umbral pesado, *ul* es el umbral liviano, *p* es la constante pesada, *l* la constante liviana.

Sustituyendo:

$$up = (1.2 * 185.3)$$

$$up = 222 \text{ (Redondeado)}$$

$$ul = (0.8 * 185.3)$$

$$ul = 148 \text{ (Redondeado)}$$

Con estos resultados, se determina que el nuevo umbral pesado será de 220 y el umbral liviano quedará en 148.

Encontrar Carga Aceptable

En este paso se seleccionarán los procesadores aceptables para recibir carga, lo cual se puede calcular restando las colas de los procesadores actuales, utilizando el nuevo umbral pesado, logrando esto con la siguiente ecuación:

$$pa = (nl_{px} \leq up - cl_{px}) \quad (6.4)$$

Donde:

nl_{px} = Carga aceptable del procesador 1

up = umbral pesado

cl_{px} = Longitud total de la Cola del procesador 1

Sustituyendo:

$$96 \leq 222 - 70$$

$$96 \leq 152$$

De manera similar, se tiene que realizar la validación por cada uno de los procesadores, obteniendo como resultado, que el procesador 4 no pueden recibir carga, siendo los procesadores 1, 2, 3, 5, 6, 7, 8, 9, 10, 11 Y 12, los procesadores aceptables para recibir la carga.

Función de Aptitud

Como fue discutido en Capítulo 5 en la sección 5.2.7, con el uso de la función de aptitud, con la ecuación 5.4, se logra obtener el fitness por cada uno de las partículas del enjambre, como se puede observar en la Tabla 6.7, que será utilizado para actualizar la posición de la partícula, así como la asignación de la velocidad inicial por cada una de las partículas.

Tabla 6.7 Posición y velocidad por partícula.

Distribución de Tareas por procesador													Posición	Velocidad
Partículas/Procesador	1	2	3	4	5	6	7	8	9	10	11	12		
1	96	97	58	128	129	88	116	100	118	108	110	105	0.186057595	0.222034213
2	85	109	124	135	140	119	77	100	99	59	98	108	0.606310708	-0.88957522
3	113	54	92	102	132	104	92	111	136	126	70	121	0.436188787	0.437353804
4	121	66	115	147	146	74	130	106	116	88	47	97	0.934038296	-0.420802993
5	109	115	149	77	111	83	89	106	122	76	67	149	0.365490082	-0.958932284
6	152	91	96	121	37	126	128	123	85	111	83	100	0.263577973	0.961839837
7	111	96	104	99	77	75	146	113	81	87	141	123	0.56365471	-0.754807816
8	133	128	111	83	100	101	50	136	108	84	101	118	0.737261913	-0.096710675
9	116	87	98	112	104	82	123	148	91	79	129	84	0.708708795	0.045512462
10	80	88	107	86	154	127	88	95	145	107	96	80	0.854975641	-0.096424725
11	76	130	93	124	99	123	124	134	116	68	43	123	0.048666878	-0.049689672
12	72	138	124	131	108	100	99	60	103	104	129	85	0.169698308	0.646801399
13	77	158	132	116	139	93	110	140	110	51	61	66	0.984298639	0.615852699
14	106	86	126	122	136	101	63	86	114	110	102	101	0.646129751	-0.367427524
15	119	134	60	117	116	105	110	110	56	80	122	124	0.117511559	-0.134215919
16	69	93	91	70	156	134	76	134	98	124	92	116	0.892566115	0.925842661
17	94	71	132	102	97	114	74	106	120	128	72	143	0.778278273	-0.327535233
18	154	102	92	99	155	69	77	83	103	89	96	134	0.392486361	-0.604118002
19	84	91	96	101	138	156	104	90	109	105	132	47	0.531107715	-0.507653238
20	98	91	122	107	59	131	121	114	93	108	114	95	0.803367971	-0.717551583
21	167	67	96	65	120	67	73	123	71	145	125	134	0.858045195	-0.44594798
22	114	105	139	113	105	95	103	54	121	98	87	119	0.614236466	0.145797881
23	135	63	109	67	108	148	84	122	126	102	67	122	0.678731511	0.793793034
24	117	93	107	119	91	110	122	77	85	90	135	107	0.696784911	0.772590592

Como se puede observar en la Figura 6.8 las partículas tienen asignada por cada una que son una posible solución al problema de optimización y que por medio de la fórmula de velocidad vista en Capítulo 4 en la sección 4.1.1, optimizando la posición de la partícula, iniciando el cálculo a partir de la mejor posición, que se obtiene en la partícula 11, seleccionada en la Tabla 6.7.

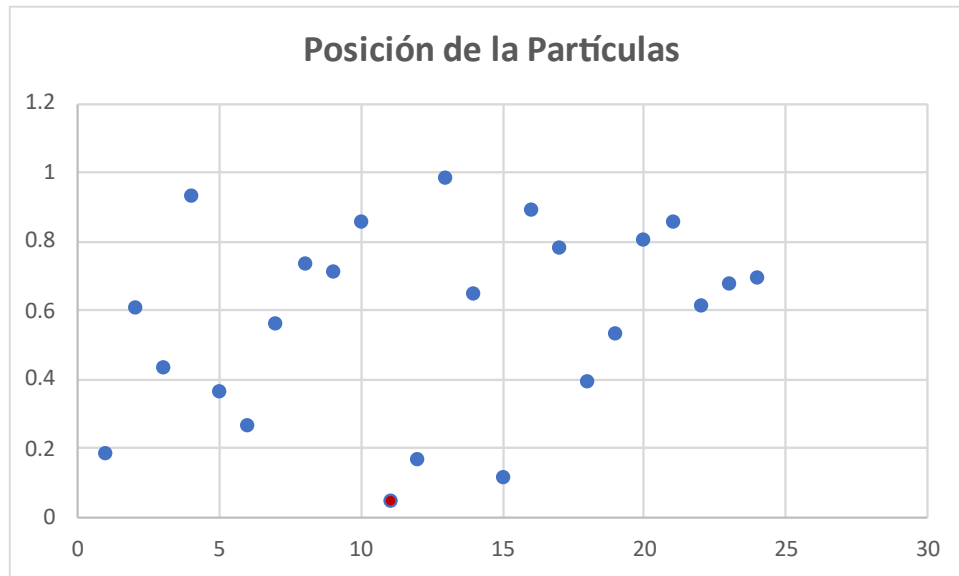


Figura 6.8 Posición de partículas.

Asignación de Tareas

Con las cargas aceptables generadas por el PSO, revisamos las tareas disponibles para cada procesador, validando si la carga total por procesador excederá la carga permitida por el umbral del sistema general que fue designado en un porcentaje global del 80%, Esto en base al porcentaje destinado como umbral pesado que fue asignado en la fórmula de 1.2 que equivale al 20% del sistema, como el procesador 4, no pueden recibir esta planificación, por exceder la carga aceptable, se tienen que sumar esas tareas en la siguiente ventana deslizando para su planificación para evitar pérdida de tareas del conjunto de datos.

Nuevo estado del sistema

Después de la asignación de tareas, el sistema tiene un cambio en la longitud total de la cola de sus procesos, donde se detecta que el procesador 4, tiene poca carga en comparación con el resto, además de que el procesador 2 no puede recibir más carga en esta ronda. Por lo tanto, se debe asignar otra lista de tareas.

Solicitando y Actualizando la ventana deslizando

Cuando todo el proceso anterior realiza, se hace la solicitud a la ventana deslizando que asigne nuevas tareas, sumando las tareas que no fueron procesadas por el procesador 2, por no poder aceptar carga, evitando tener pérdida de tareas por nuestro sistema, de esta forma la nueva ventana de tareas quedaría como se puede observa en la Tabla 6.8, con este nuevo conjunto de tareas para ser planificada, se tiene que regresar a los pasos

anteriores y repetirlos nuevamente hasta terminar todo el conjunto de datos recolectado, como se.

Tabla 6.8 Nuevo contenido de Ventana Deslizante

Tareas sin asignar	Nuevas tareas	Tareas para planificar
73(42),85(5),97(35), 69(46)	20,2,36,13,5,32,46,8,20,27,4,24,8,37,36,27,16,21,17,36, 36,38,11,46,7,42,21,24,41,23,18,17,30,20,27,45,33,33,3 2,18,27,16 17,32,9,15,14,33	42,5,35,46 ,20,2,36,13,5,32,46,8,20,27,4,24,8, 37,36,27,16,21,17,36,36,38,11,46,7,42,21,24, 41,23,18,17,30,20,27,45,33,33,32,18,27,16 17,32,9,15,14,33

De acuerdo con los resultados en esta simulación se logra observar que el uso de la calibración de carga junto con los umbrales adecuados y el tamaño correcto de iteraciones y posibles soluciones se obtienen resultados cercanos al óptimo, aunque no podemos comparar directamente nuestros resultados en tiempo de preprocesamiento, podemos hacerlo en términos de estrategia de planificación de tareas como la de Zomaya et al. [124], donde podemos observar que el uso correcto de generaciones, poblaciones y una programación de carga flexible del 20% equivalente en los umbrales $H=1.2$ y $L=0.8$, puede lograr aumentar significativamente la utilización de los procesadores. Aunque el uso de la metaheurísticas es completamente diferente se puede estar de acuerdo, que la correcta elección del tamaño de población sigue siendo una tarea de suma importante para este tipo de problemas de planificación de tareas.

6.5 Experimento de Planificación de Tareas con ACO

El proceso de planificación de tareas utilizando ACO incluye diversas técnicas que se detallan a continuación. Estas técnicas demuestran cómo este enfoque optimiza los recursos del sistema y mejora su rendimiento al reducir los tiempos de espera. A partir de la primera lista de tareas generada mediante la técnica de ventana deslizante, se crea una "ventana" de tareas que se transferirá y procesará a lo largo de las diferentes etapas del algoritmo ACO.

Técnica de permutación de tareas: Se refiere a la técnica de reordenar tareas entregadas de la primera "ventana" de la técnica de la ventana deslizante, de esta manera buscar la optimización del sistema, generando 24 hormigas a partir de esta ventana, como se puede observar en la Figura 6.9.

Permutación ACO																																															
22	18	46	29	44	19	10	2	22	40	43	12	4	35	41	46	37	16	2	27	41	26	4	41	32	22	7	9	29	33	7	34	22	37	45	13	8	6	8	19	2	46	42	9	5	42	35	9
40	6	19	35	29	43	4	8	32	7	41	27	44	9	33	12	26	42	41	13	29	19	46	37	8	34	46	22	9	22	5	2	35	9	42	2	46	10	37	45	22	7	18	4	22	41	16	2
9	8	22	8	5	37	46	22	13	4	37	45	34	2	33	35	26	9	18	2	44	29	41	42	7	16	4	35	29	22	19	42	43	7	46	2	12	40	22	9	46	19	10	27	32	41	6	41
32	22	2	22	12	41	29	43	42	2	45	5	34	44	46	22	6	4	37	10	8	26	19	9	35	7	37	42	22	41	16	2	9	33	13	8	41	4	27	46	7	9	46	29	35	19	40	18
13	46	34	29	22	10	6	22	4	44	32	22	42	2	37	4	40	16	41	33	35	37	18	9	43	41	26	8	35	45	46	7	7	19	29	8	12	2	19	27	9	9	41	5	46	2	42	22
46	12	19	26	33	44	18	9	7	27	19	32	10	46	43	41	34	13	40	45	2	2	9	8	4	7	42	35	9	41	5	16	29	37	22	22	2	4	35	46	29	2	42	41	22	37	8	6
22	2	16	10	41	45	12	42	34	9	19	22	18	42	46	40	44	7	41	35	6	4	19	37	32	22	9	5	9	29	26	29	33	8	46	46	37	13	2	22	7	27	43	4	2	35	8	41
41	12	42	7	6	19	9	19	35	32	44	35	4	13	46	16	33	27	46	22	5	9	22	10	41	9	7	2	29	34	45	26	43	8	2	8	2	37	18	22	46	22	4	41	42	29	37	40
13	35	40	19	22	26	32	33	27	22	43	19	9	8	2	12	37	5	29	35	41	34	44	22	46	41	16	2	9	4	8	18	7	6	2	22	45	42	46	41	4	10	9	42	29	37	7	46
2	27	9	9	35	32	22	22	40	37	42	5	43	37	46	46	33	41	26	34	7	12	41	18	44	42	46	19	16	8	9	19	22	29	4	4	41	2	8	29	35	10	22	7	13	2	6	45
42	41	35	13	7	18	9	29	45	9	33	32	43	22	6	44	2	27	7	46	12	35	2	26	19	8	29	22	22	2	4	9	5	41	46	34	46	22	19	8	42	37	16	37	41	4	10	40
34	44	2	7	27	37	41	7	9	4	42	29	41	43	45	22	35	32	9	8	19	41	12	46	8	40	37	42	2	16	46	5	46	6	22	33	29	2	10	4	9	19	18	22	13	26	22	
33	35	42	13	29	27	2	37	37	9	10	29	46	18	5	9	4	41	2	44	8	46	7	4	22	6	45	16	19	46	9	32	42	7	40	2	43	8	22	41	41	12	35	22	34	19	22	26
22	6	35	33	46	37	12	37	22	40	46	9	44	16	8	26	32	34	35	22	27	46	41	4	5	29	29	42	7	41	2	18	7	8	2	9	10	13	2	4	42	45	43	19	22	9	19	41
44	35	29	41	41	2	16	4	9	32	22	45	29	19	37	7	34	27	12	42	5	43	46	2	22	41	9	4	8	22	33	10	7	46	40	35	9	13	18	22	42	19	2	37	46	6	26	8
5	4	10	46	37	29	4	9	40	44	35	7	34	9	32	2	35	26	22	41	7	42	22	22	43	46	2	37	45	29	42	19	12	41	19	33	46	9	27	8	41	16	8	22	2	13	6	18
13	9	2	2	9	22	22	42	22	37	37	4	2	16	29	42	4	6	33	8	44	19	46	29	27	41	26	7	40	22	46	32	7	45	18	19	9	35	5	41	46	12	10	35	8	41	43	34
9	41	18	6	16	8	9	4	19	12	42	26	22	7	41	37	46	29	33	22	40	2	19	46	34	9	22	2	5	42	10	35	2	8	43	45	46	37	4	13	41	44	32	7	35	29	27	22
22	18	33	29	9	42	19	41	16	22	42	46	8	29	32	41	46	40	12	37	9	7	5	6	7	35	35	9	8	10	26	41	34	46	27	2	22	37	13	45	22	19	4	43	2	4	2	44
2	4	18	35	37	10	19	33	44	40	16	9	2	42	5	43	12	13	29	22	46	41	46	4	27	9	41	41	8	45	34	8	9	42	7	22	22	19	7	32	6	2	46	26	22	35	29	37
33	22	5	45	29	40	2	43	42	16	46	32	22	7	46	35	7	8	19	19	9	2	41	18	26	27	44	13	46	35	29	41	41	37	4	22	2	37	12	9	9	22	42	34	10	4	8	6
9	5	46	41	9	19	29	46	2	4	44	12	34	22	41	2	40	46	42	22	42	19	16	22	35	8	26	41	7	2	22	33	4	9	7	27	32	6	29	45	8	18	37	10	43	35	37	13
19	35	9	26	46	33	46	9	41	22	22	42	4	29	6	44	2	7	42	37	13	46	43	41	2	7	18	10	41	2	9	29	5	34	4	19	12	45	8	40	37	32	22	16	22	27	35	8
18	41	37	5	2	46	43	4	8	19	2	26	13	8	32	29	6	9	46	40	22	7	37	35	46	41	7	42	42	9	12	10	29	45	4	22	22	16	19	34	2	22	41	9	27	35	44	33

Figura 6.9 Permutación de hormigas.

Estado Actual de los Procesadores

En este paso, se recolecta el estado actual de los procesadores, para visualizar que tareas se encuentran ejecutando en el sistema por cada uno de los procesadores, como se visualiza en la Tabla 6.9.

Tabla 6.9 Estado del Actual de los Procesadores.

Núm. Procesador	Tareas Actuales	Cola del Procesador	Longitud total de la Cola
1	9(3)	1(17),2(30),3(40),4(32)	122
2	10(20)	5(44),6(32),7(34),8(24)	154
3	4(19)	9(4),10(8),11(42),12(6)	79
4	11(24)	13(34),14(46),15(20),16(17)	141
5	20(8)	17(34),18(35),19(34),20(13)	124
6	18(15)	21(30),22(26),23(41),24(8)	120
7	3(6)	25(4),26(29),27(42),28(35)	116
8	2(4)	29(24),30(5),31(38),32(43)	114
9	5(12)	33(22),34(11),35(17),36(6)	68
10	8(12)	37(35),38(2),39(3),40(40)	92
11	30(3)	41(40),42(21),43(32),44(29)	125
12	22(12)	45(32),46(35),47(40),48(17)	136

Nuevas tareas que llegan para ser planificadas

En este paso se detalla la llegada del nuevo conjunto de tareas (tweets) [22,18,46,29,44,19,10,2,22,40,43,12,4,35,41,46,37,16,2,27,41,26,4,41,32,22,7,9,29,33,7,34,22,37,45,13,8,6,8,19,2,46,42,9,5,42,35,9] despues de haber sido permutadas en con la técnica de permutación de tareas.

Asignación de la tareas por procesador.

Basándose en el conjunto de nuevas tareas que arribaron al sistema, se ejecuta la asignación de tareas, acorde al arribo de cada una de ellas, ejemplo, la tarea 10, es asignada en el procesador 1, la tarea 8 es asignada al procesador 2 y así hasta terminar con el procesador 12, retornando al procesador 1, asignándole la tarea 12, hasta posicionar todas las nuevas tareas en los procesadores como se puede observar en la Tabla 6.10. En la columna “Nueva Carga” se establecen las nuevas tareas que serán agregadas a los procesadores.

Tabla 6.10 Tareas asignadas en los procesadores

Núm. Procesador	Nuevas tareas asignadas	Nueva Carga
1	70(22),82(4),94(32),66(8)	66
2	71(18),83(35),95(22),67(6)	81
3	72(46),84(41),96(7),68(8)	102
4	73(29),85(46),97(9),69(19)	103
5	74(44),86(37),98(29),54(2)	112
6	75(19),87(16),99(33),55(46)	114
7	76(10),88(2),60(7),56(42)	61
8	77(2),89(27),61(34),57(9)	72
9	78(22),90(41),62(22),58(5)	90
10	79(40),91(26),63(37),59(42)	145
11	80(43),92(4),64(45),23(35)	127
12	81(12),93(41),65(13),25(9)	75

Determinar la carga promedio de los procesadores

El siguiente para es determinar los nuevos umbrales basados en la carga actual del sistema y las nuevas tareas que arribaron y se planificaron para que se programe a integrarse al sistema, para obtenerlo, utilizaremos la siguiente fórmula:

$$lave = ((\sum cl + \sum nl) / n) \quad (6.1)$$

Donde, se obtendrá el porcentaje de la carga (*lave*), realizando la sumatoria de la carga actual (*cl*) y la nueva carga (*nl*) dividida entre el número de procesadores (*n*).

$$cl = 122 + 154 + 79 + 141 + 124 + 120 + 116 + 114 + 68 + 92 + 125 + 136$$

$$nl = 66 + 81 + 102 + 103 + 112 + 114 + 61 + 72 + 90 + 145 + 127 + 75$$

$$n = 12$$

Sustituyendo:

$$lave = (1391 + 1148) / 12.$$

$$lave = 211.5$$

Umbrales Pesados y Livianos

Con el resultado del porcentaje de la carga (*lave*), podemos calcular los umbrales con las siguientes ecuaciones:

$$up = p * lave \quad (6.2)$$

$$ul = l * lave \quad (6.3)$$

Donde, up es el umbral pesado, ul es el umbral liviano, p es la constante pesada, l la constante liviana.

Sustituyendo:

$$up = (1.2 * 211.5)$$

$$up = 254 \text{ (Redondeado)}$$

$$ul = (0.8 * 211.5)$$

$$ul = 169 \text{ (Redondeado)}$$

Con estos resultados, se determina que el nuevo umbral pesado será de 254 y el umbral liviano quedará en 169.

Encontrar Carga Aceptable

En este paso se seleccionarán los procesadores aceptables para recibir carga, lo cual se puede calcular restando las colas de los procesadores actuales, utilizando el nuevo umbral pesado, logrando esto con la siguiente ecuación:

$$pa = (nl_{px} \leq up - cl_{px}) \quad (6.4)$$

Donde:

nl_{px} = Carga aceptable del procesador 1

up = umbral pesado

cl_{px} = Longitud total de la Cola del procesador 1

Sustituyendo:

$$66 \leq 254 - 122$$

$$66 \leq 132$$

De manera similar, se tiene que realizar la validación por cada uno de los procesadores, obteniendo como resultado, que todos los procesadores 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 Y 12, pueden recibir carga para su procesamiento.

Función de Aptitud

Como fue discutido en Capítulo 5 en la sección 5.2.7, con el uso de la función de aptitud, con la ecuación 5.4, se logra obtener el fitness por cada uno de las hormigas de la colonia, como se puede observar en la Tabla 6.11, que será utilizado para inicializar la visibilidad y la feromona por cada una de las hormigas.

Tabla 6.11 Visibilidad y Feromonas por hormiga.

Distribución de Tareas por procesador													Fitness f(x)	Visibilidad [n(e)]	Feromonas [t(n)]
Hormigas/Procesador	1	2	3	4	5	6	7	8	9	10	11	12			
1	66	81	102	103	112	114	61	72	90	145	127	75	0.055921053	17.88235294	0.1
2	138	59	135	114	86	114	68	27	118	76	145	68	0.050898204	19.64705882	0.1

Al finalizar la primera iteración del ACO se debe realizar la evaporación de los recorridos de la hormiga, como se puede observar en la Tabla 6.12, el mejor camino fue obtenido por la hormiga 24, ya que tuvo la mejor probabilidad.

Tabla 6.12 Actualización de la probabilidad de la hormiga.

<i>Hormigas/Procesador</i>	<i>t(n)(e)</i>	<i>Probabilidad Pxy</i>
1	0.049299192	0.001202419
2	0.593827938	0.014483608
3	-0.100673249	-0.002455445
4	0.273938623	0.00668143
5	1.543806642	0.037653821
6	-0.623351704	-0.0152037
7	1.058912684	0.025827139
8	0.145027437	0.003537255
9	-0.066179155	-0.001614126
10	0.149507827	0.003646532
11	0.079513566	0.001939355
12	-0.616661534	-0.015040525
13	-0.207880445	-0.005070255
14	0.259027767	0.00631775
15	0.112632981	0.002747146
16	0.080224496	0.001956695
17	0.07889832	0.001924349
18	0.510481515	0.012450769
19	0.686680313	0.0167483
20	1.485480888	0.036231241
21	1.12207459	0.027367673
22	-0.34559316	-0.008429101
23	-1.251466213	-0.030523566
24	-0.621142965	-0.015149828

Asignación de Tareas

Con las cargas aceptables generadas por el ACO, revisamos las tareas disponibles para cada procesador, validando si la carga total por procesador excederá la carga permitida por el umbral del sistema general que fue designado en un porcentaje global del 80%, Esto en base al porcentaje destinado como umbral pesado que fue asignado en la fórmula de 1.2 que equivale al 20% del sistema.

Nuevo estado del sistema

Después de la asignación de tareas, el sistema tiene un cambio en la longitud total de la cola de sus procesos, donde se detecta que todos los procesadores pueden aceptar carga.

Solicitando y Actualizando la ventana deslizante

Cuando todo el proceso anterior se realiza, se hace la solicitud a la ventana deslizante que asigne nuevas tareas.

Tabla 6.13 Nuevo contenido de Ventana Deslizante

Tareas sin asignar	Nuevas tareas	Tareas para planificar
0	38,27,40,17,25,46,29,14,29,2,36,44,43,27,25,21,24,44,3,6,8,5,18,42,7,45,27,39,31,9,33,22,3,40,22,34,21,13,44,25,9,30,35,36,34,18,40,24,30	38,27,40,17,25,46,29,14,29,2,36,44,43,27,25,21,24,44,3,6,8,5,18,42,7,45,27,39,31,9,33,223,40,22,34,21,13,44,25,9,30,35,36,34,18,40,24,30

De acuerdo con los resultados en esta simulación se logra observar que el uso de la calibración de carga junto con los umbrales adecuados y el tamaño correcto de iteraciones y posibles soluciones se obtienen resultados cercanos al óptimo, aunque no podemos comparar directamente nuestros resultados en tiempo de preprocesamiento, podemos hacerlo en términos de estrategia de planificación de tareas como la de Zomaya et al. [124], donde podemos observar que el uso correcto de generaciones, poblaciones y una programación de carga flexible del 20% equivalente en los umbrales $H=1.2$ y $L=0.8$, puede lograr aumentar significativamente la utilización de los procesadores. Aunque el uso de la metaheurísticas es completamente diferente se puede estar de acuerdo, que la correcta elección del tamaño de población sigue siendo una tarea de suma importante para este tipo de problemas de planificación de tareas.

6.6 Precisión de predicción en clasificadores

Con la metodología propuesta se logró una colección de más de un millón de tweets, dando como resultado un conjunto de datos con el análisis de sentimiento e identificación de palabras y frases suicidas utilizando expresiones regulares que mantienen una relación con los autores [120] [112], los resultados obtenidos en estas investigaciones determinaron la necesidad de tener líneas de ayuda a personas que estén en peligro y tal vez en riesgo de suicidio a largo plazo, buscando minimizar los falsos positivos y maximizando los verdaderos positivos para evitar de esta forma sobrecargar a los servicios de emergencia con falsas alarmas. Basado en las estrategias de los autores se puede visualizar que nuestra investigación no guarda una relación con ellos en la recolección de los datos ya que estos fueron recolectados de datos anónimos de foros, Internet, blogs y micro blogs sobre temas de suicidio para lograr extraer las frases y palabras necesarias para el filtro utilizado en la búsqueda en Twitter, en contraste a los utilizados por esta investigación que fueron extraídos de la literatura, como ciencias de la psicología con datos clínicos y la ciencia de tecnología con datos no clínicos, que fueron detallados en el Capítulo 5, sección 5.2.3,

para ser utilizados en el filtro de extracción de API de Twitter. Donde sí podemos visualizar que guarda una relación es con el uso de diferentes técnicas de clasificación, como la del suicidio, frivolidad y análisis del sentimiento, con la diferencia que los autores utilizan un conjunto de datos por cada una de las clasificaciones en contraste con el nuestro, ya que para optimizar los tiempos, la generación del conjunto de datos fue realizado en un solo conjunto de datos con las 3 técnicas de clasificación. Como se puede observar en la Tabla 6.14, sumada a la técnica de eliminación de ruido, se logró una reducción considerable del 30% del total de tweets recolectados, que como recordamos son más de 1,000,000, y que será utilizada por los algoritmos seleccionados en esta investigación.

Tabla 6.14 Conjunto de Datos preprocesado con Modelo propuesto

Conjunto de Datos						
Suicidio		Frivolidad		Sentimiento		
Suicidio (-1)	No Suicidio (1)	Frívola (-5)	No Frívola (5)	Negativo (-10)	Positivo (10)	Neutral (0)
506,653	207,734	250,210	464,177	313,115	205,532	195,740

Con estos datos se observa que el 29% del total de datos fueron clasificados como “no suicidio” (1), aun cuando en nuestra metodología se realizó el filtro de palabras y frases relacionadas con el tema suicidio, obteniendo similares resultados con la clasificación de sentimientos con el 28% que clasificado como positivo (10) y donde el 27% se consideró neutral (0), lo cual no es inusual al ser la clase predeterminada en el análisis de sentimiento, el 44% restante se logró clasificar como negativo (-10), por el contrario, el 71% de los tweets fueron clasificados como suicidio, lo que sugiere que la técnica de análisis de sentimiento puede tener áreas de oportunidad al clasificar erróneamente el 27% de los datos, en comparación con la técnica de expresión regular, que detecta más características de ideación suicida que podrían tener el contenido del tweet. En la Tabla 6.15, se puede visualizar las características extraídas por nuestro modelo que tienen relación con la ideación suicida y no solamente con el filtrando palabras y frases recogidas de la literatura en la primera etapa de nuestra metodología, optamos por utilizar una estrategia no supervisada como el modelado de temas mediante Análisis Latente de Dirichlet (LDA), que es un método probabilístico para extraer temas de documentos, asumiendo que cada documento tiene uno o más temas con diferentes distribuciones, configurando el método con 3 componentes con las 20 características más dominantes que fueron asociadas con los siguientes temas como el lenguaje suicida, la ideación suicida y la búsqueda de ayuda.

Tabla 6.15 Validación del Modelo con LDA

Lenguaje Suicida	Ideación Suicida	Búsqueda de Ayuda
suicidio, gente, triste, saber, ir, vida, alguien, cosa, no, nunca, decir, todavía, algo, llorar, mierda, hombre, joder, derecho, mal, niño, mujer, escuela, odio, semana, desde, mundo, gran, importante,	Ansiedad, como, salud, mental, conseguir, depresión, hacer, sentir, necesidad, muerte, mirar, manera, siempre, nueva, dijo, atrás, cuidado, venir, favor, mejor, cada, perra, reloj, hecho, real, causa, sentimiento historia	triste estrés tiempo realmente tiene incluso amigo quiere hoy amor vivo año no puede pensar depresión suficiente hija café descrito noticias ayuda tomar eso persona también dios familia amable

comprometerse, amigo, primero, puede, razón, chica, negro romper	dio colegio tema pensamiento hablar recordar deseo denunciante nota nada	mejor parado informe dar último oír bien día lejos.
--	--	---

Con el conjunto de datos recolectado y preprocesado utilizando el algoritmo bioinspirado con las 140 características extraídas con word2vec como variables predictoras y la variable a predecir generadas con las técnicas de clasificación explicadas en la sección anterior, se procedió a utilizar las técnicas de validación cruzada vistas en el Capítulo 5, sección 5.2.6.

Por otra parte, Word2Vec utiliza una red neuronal, típicamente una red neuronal de una sola capa oculta. El objetivo del entrenamiento es ajustar los pesos de la red de manera que la representación vectorial de una palabra sea similar a la de las palabras que aparecen en su contexto. Una vez que el modelo está entrenado, cada palabra en el vocabulario tiene asociado un vector de números reales. Estos vectores representan la semántica de las palabras en un espacio de características. El número adecuado de dimensiones para utilizar en Word2vec varía según el tamaño y la complejidad del corpus de texto, así como la tarea específica que se esté abordando. El hiperparámetro `vector_size` en Word2vec especifica la dimensionalidad de los vectores de palabras que el modelo va a aprender durante el entrenamiento. En otras palabras, determina cuántas dimensiones se asignan a cada palabra en el espacio vectorial. En la literatura no se encuentra una recomendación universalmente aceptada ni una técnica específica preferida, pero generalmente se utilizan dimensiones en el rango de 100 a 300. Algunos estudios han empleado dimensiones que varían de 100 a 1000. Sin embargo, es crucial considerar la eficiencia computacional, ya que dimensiones más altas incrementan los costos de procesamiento. Por otro lado, un valor más bajo puede resultar en representaciones vectoriales más simples, pero puede perder información semántica. Experimentamos con diferentes valores de 100, 150 y 300 dimensiones para evaluar el rendimiento del modelo. La diferencia en eficiencia computacional entre 100 y 150 dimensiones es mínima en comparación con la diferencia entre 100 y 300 dimensiones, lo que afecta significativamente nuestro objetivo principal de mejorar el tiempo computacional. En la Figura 6.11, se presentan los tiempos obtenidos en los experimentos realizados con los valores mencionados, empleamos la técnica Skip-gram, que fue la arquitectura que nos proporcionó mejores tiempos.

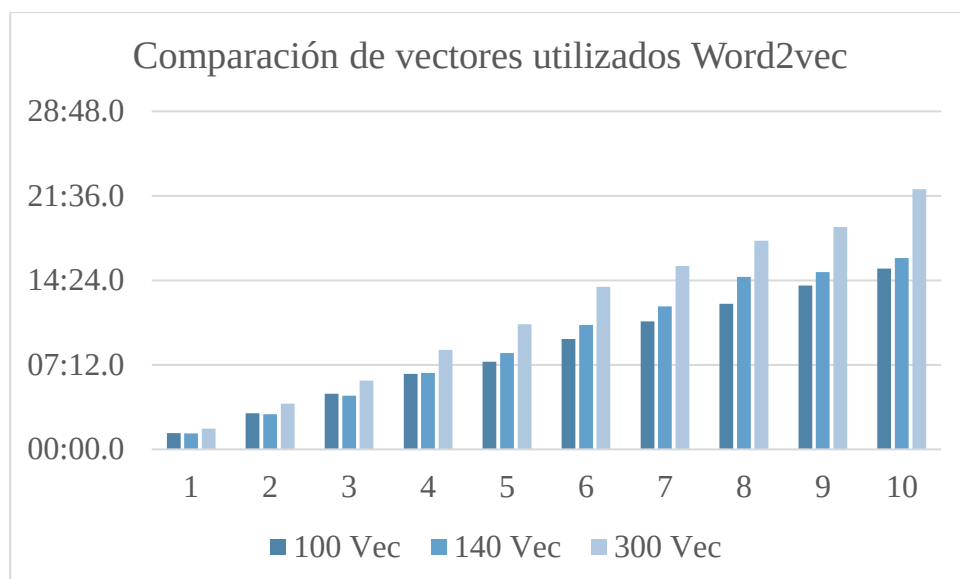


Figura 6.11 Comparación de número de vectores en Word2vec.

En la Tabla 6.16 se observan los resultados obtenidos, donde la técnica Hold-Out sobre clasificadores como KNN, RF y MLP logra buenos resultados, en comparación con LR y SVM, donde se puede ver podemos ver una mejora en la técnica Leave One Out con los algoritmos KNN y MLP pero una reducción con el LR, notando en los resultados obtenidos en el K-Fold que se tuvo una constante del 80% en los algoritmos SVM, KNN, RF y MLP, y solo en el LR por debajo de este porcentaje.

Tabla 6.16 Algoritmos Experimentales y Técnicas de Validación

Algoritmo	Técnica de Validación					
	Hold-Out		Leave-One-Out		K-Fold	
	Predicción	Factor de Error	Predicción	Factor de Error	Predicción	Factor de Error
Logistic Regression	0.7527	0.2473	0.7427	0.2573	0.7513	0.2487
Support Vector Machines	0.7636	0.2364	0.7601	0.2399	0.8754	0.1246
K-Nearest Neighbors	0.8905	0.1095	0.9145	0.0855	0.8864	0.1136
Random Forest	0.9323	0.0677	0.8188	0.1812	0.8559	0.1441
Multi-layer Perceptron	0.8687	0.1313	0.9075	0.0925	0.8029	0.1971

La extracción de características del suicidio con la técnica de expresiones regulares tiene una predicción aceptable con el Algoritmo Random Forest, obtenido el 0.9323 con un factor de error del 0.0677 con la técnica Hold Out. De acuerdo con los resultados obtenidos por Burnap et al. [112] logro la extracción de características del suicidio con la estrategia de expresiones regulares, evaluando el conjunto de datos con técnicas de validación cruzada de K-Fold, que fue entrenando con K=10, entrenada iterativamente con el 90% de los datos y con 10% de datos de pruebas obteniendo 0.6570 con el algoritmo SVM, que en comparación con nuestros resultados con la misma técnica de K-Fold y SVM obtuvimos una predicción del 0.8754 que equivale a 0.2184 de incremento. Donde no se logró una mejora significativa fue con los resultados presentados por Ahujaa et al [22]

donde obtuvo una predicción del 0.8571, con la misma técnica de validación cruzada y clasificador K-Fold y SVM respectivamente, obteniendo solamente una mejora del 0.0183. A diferencia con los autores referidos, el conjunto de datos utilizado en nuestra investigación para la validación con las diferentes técnicas de validación cruzada que fue de 700 mil tweets en comparación con el utilizado por ellos que fue una muestra muy pequeña, aun cuando se describe que se recolectaron un alto nivel de contenido por arriba del millón de tweets.

6.7 Optimización de tiempo

La evaluación de los modelos es fundamental para determinar su precisión, rendimiento, eficiencia y otras métricas clave en diversas situaciones. Otra estrategia común consiste en comparar varios algoritmos para identificar cuál es más eficiente bajo determinadas condiciones. Además, la calibración de hiperparámetros resulta crucial para encontrar los valores óptimos que maximicen el desempeño de los modelos seleccionados. Esta estrategia fue aplicada y validada con los algoritmos bioinspirados utilizados a lo largo de la investigación. El GA fue calibrado como se muestra en la Tabla 6.17 y logra obtener una mejora significativa contra el sistema secuencial tradicional como se puede observar en la Figura 6.12.

Tabla 6.17 Calibración de GA

GA	
Parámetros	Valores
Número de Generaciones (t)	10,20,30,40 y 50
Tamaño de Población (te)	24, 32 y 48
Probabilidad de Cruzamiento	0.99,0.95,0.90,0.85 y 0.80
Probabilidad de Mutación	0.1 y 0.5
Selección por ruleta	
Operador de Cruzamiento de ciclos (CX)	
Calibración final utilizada: G=10,TP=24 C=0.95 y M=0.5.	

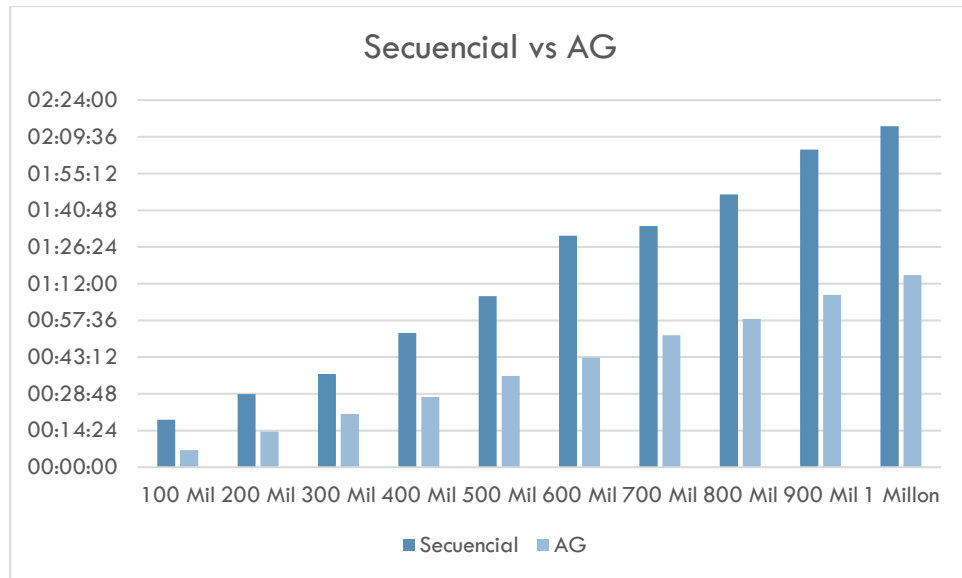


Figura 6.12 Comparación secuencial vs AG.

El PSO fue calibrado como se muestra en la Tabla 6.18 y logra obtener una mejora significativa contra el sistema secuencial tradicional como se puede observar en la Figura 6.13

Tabla 6.18 Calibración del PSO

PSO	
Parámetros	Valores
Número de Generaciones (t)	10,20,30,40 y 50
Tamaño de Enjambre (te)	24, 32 y 48
Inercia	
Factor de Aceleración (c1,c2)	1,2
Perturbaciones aleatorias (r1, r2)	
Calibración final utilizada: t=10, te=24, w=1, c1=1, c2=1,r1= rand (0,1), r2= rand (0,1)	

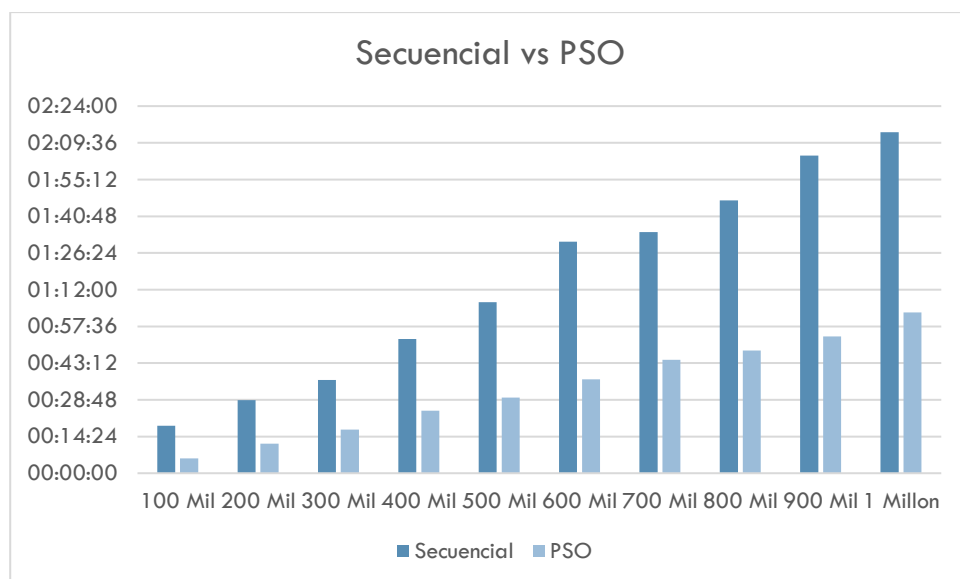


Figura 6.13 Comparación secuencial vs PSO.

El ACO fue calibrado como se muestra en la Tabla 6.19 y logra obtener una mejora significativa contra el sistema secuencial tradicional como se puede observar en la Figura 6.14

Tabla 6.19 Calibración del ACO

ACO	
Parámetros	Valores
Número de Generaciones (t)	10,20,30,40 y 50
Tamaño de Colmena (te)	24, 32 y 48
ALPHA	1,0.5
BETA	1,0.5
rho	0.5
tau	0.5
Calibración final utilizada: t=10, te=24, alpha=1,beta=1, rho=.5, tau=.5	

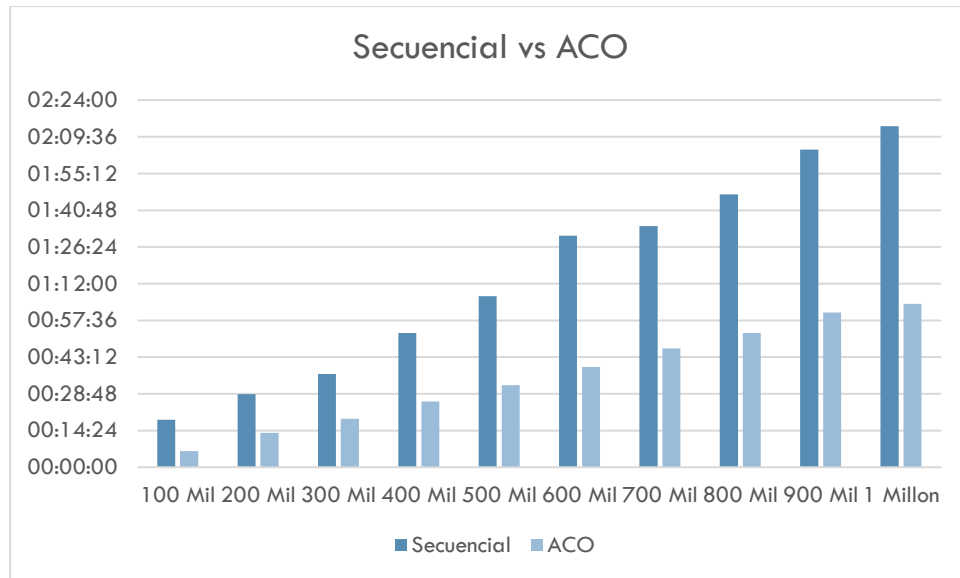


Figura 6.14 Comparación secuencial vs ACO.

En comparativas anteriores las técnicas de algoritmos bioinspirados, superan por mucho a los sistemas secuenciales tradicionales, yaque no optimiza los tiempos del preprocesamiento. En la Figura 6.15, se hace la comparativa entre los algoritmos bioinspirados utilizados en esta investigacion, como el AG, PSO y ACO, donde se puede concluir que el algoritmo que nos da mejores resultados en este tipo de problemas de optimizacion de tiempos en la planeación de tareas para detectar el suicidio es el PSO.

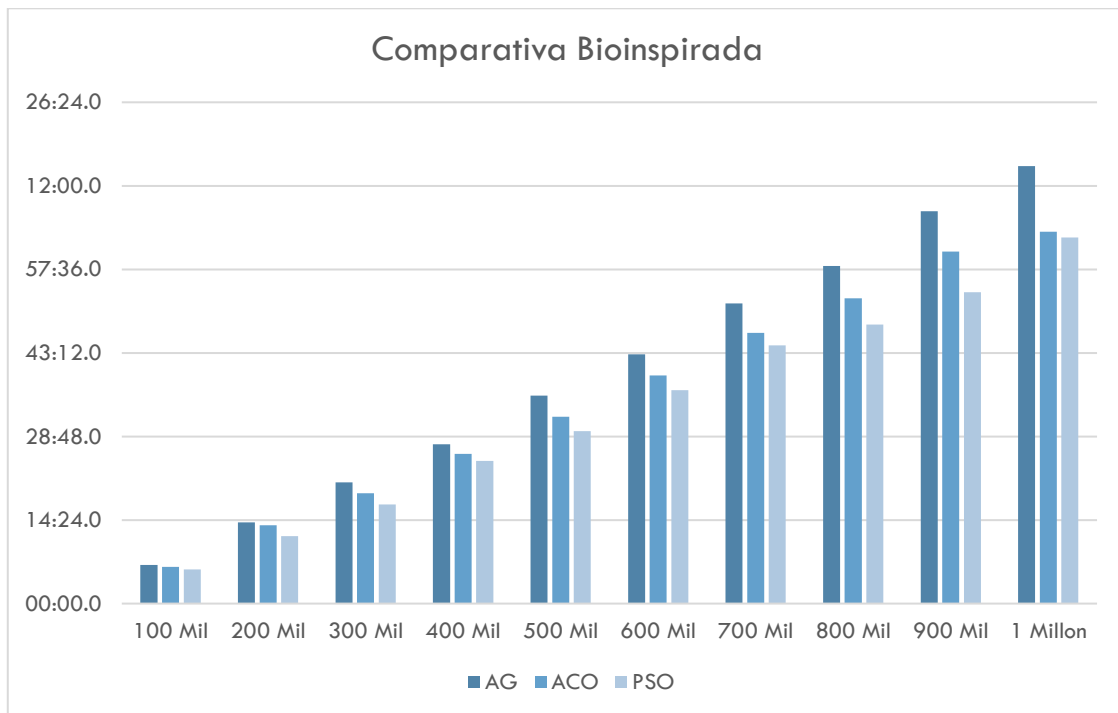


Figura 6.15 Comparación Bioinspirada.

Se realizó otra evaluación de optimización de tiempos, haciendo una comparativa con los algoritmos de planificación tradicionales como Round Robin (RR), First-Come, First-Served(FCFS) y Shortest Job First(SJF), que son eficientes para la gestión de planificación de tareas, como se puede observar en la Figura 6.16, y se identifica que los algoritmos bioinspirados, siguen logrando un mejor resultado.

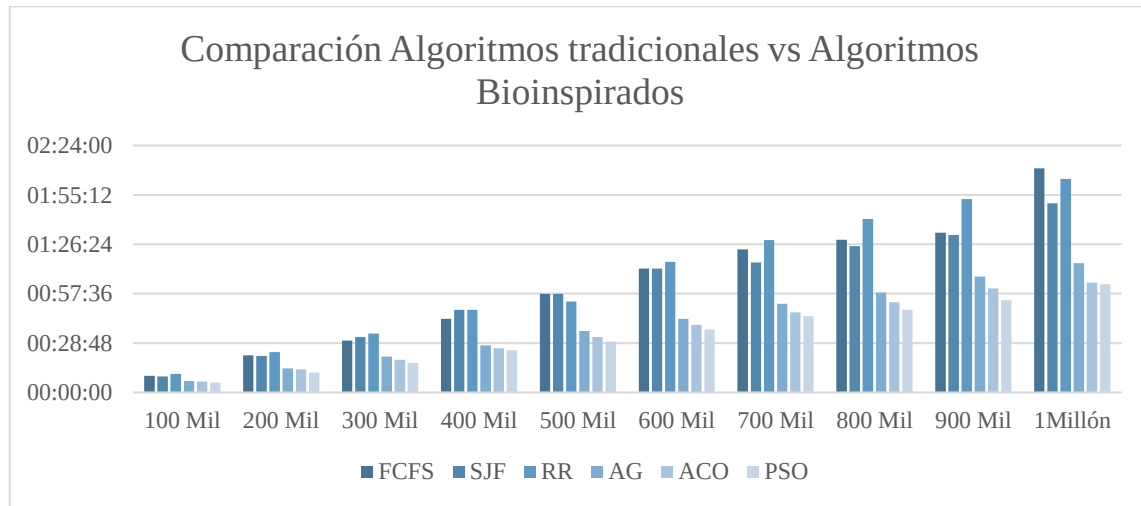


Figura 6.16 Comparación Algoritmos tradicionales vs Algoritmos Bioinspirados.

Cabe recalcar que los algoritmos tradicionales no están diseñados para la planificación adaptativa, ya que regularmente este tipo de algoritmos son estáticos, es decir no evalúan la carga del sistema en tiempo de ejecución, generando altos costos de computación en tareas de PLN.

En la Tabla 6.20 se puede observar que el millón de tweets recopilados, fueron distribuidos en 10 conjuntos de datos repartidos en una forma incremental de 100 mil por cada ciclo, para poder realizar y detectar entre cada ciclo como la optimización global del tiempo se puede visualizar conforme se va incrementando el número de tweets enviados a procesar en una forma secuencial tradicional y con estrategia bioinspirada. Logrando con la estrategia bioinspirada una optimización global del tiempo en el preprocesamiento de los datos del 134%, lo cual demuestra que el objetivo principal de nuestra investigación cumple con su cometido, ya que la planificación de las cargas generada por el algoritmo bioinspirado seleccionado a logrado que los procesadores reduzcan significativamente el tiempo de ejecución.

Tabla 6.20 Mejora de Tiempo Secuencial vs Bioinspirado.

Ciclos	Cantidad de Tweets	Tiempo Secuencial	Tiempo PSO	% Mejora de Tiempo
1	100,000	00:18:36	00:05:52	217%
2	200,000	00:28:38	00:11:37	146%
3	300,000	00:36:31	00:17:06	114%
4	400,000	00:52:38	00:24:34	114%
5	500,000	01:07:04	00:29:43	126%
6	600,000	01:30:52	00:36:47	147%
7	700,000	01:34:33	00:44:30	112%
8	800,000	01:47:05	00:48:06	123%
9	900,000	02:04:35	00:53:40	132%
10	1,000,000	02:13:44	01:03:07	112%

El proceso secuencial comenzó con la lectura del número total de tweets con información relevante que el usuario compartió en Twitter, cada tweet es procesado utilizando los diferentes métodos que se explicaron en la metodología del Capítulo 5. En Tabla 6.20 se observa que un número cada vez mayor de tweets tiene un impacto significativo en el tiempo de procesamiento. Un claro ejemplo en el primer ciclo, donde se puede observar que con 100.000 tweets, tomó un tiempo de 0:18:36 minutos para preprocesar esa cantidad de tweets con la arquitectura de secuencia tradicional utilizando toda la capacidad del sistema como procesador y memoria, en comparación con la planificación realizada con algoritmo bioinspirado, que mejoró significativamente el tiempo a 0:05:52 minutos para la misma cantidad de tweets, como resultado se obtiene una mejora relevante del 217% en el tiempo de procesamiento, tan solo en el primer ciclo, si revisamos los 10 ciclos completos, se puede observar que se va obteniendo una mejora entre cada ciclo.

Los ejemplos mostrados en las Figura 6.17 y Figura 6.18, refleja como la técnica bioinspirada planifica las cargas en los procesadores, donde se visualiza que los 12 procesadores están trabajando de una forma más equilibrada, a diferencia del sistema secuencial tradicional, donde se puede ver que el sistema considera que la carga puede ser “atendida” por un solo procesador dejando el resto en un estado casi en ocio como es el caso el procesador 11 y 12.

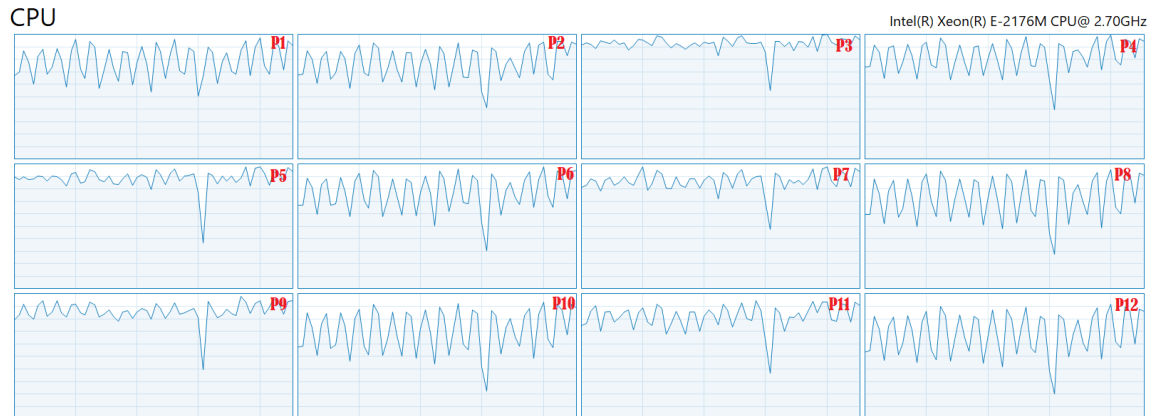


Figura 6.17 Rendimiento de los Procesadores con estrategia bioinspirada

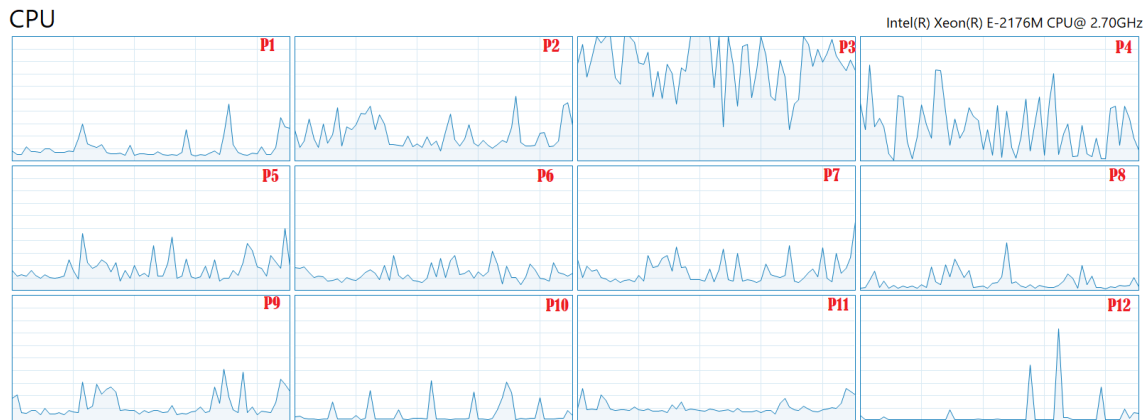


Figura 6.18 Rendimiento de los Procesadores en Procesamiento Secuencial

6.8 Conclusiones del capítulo

En este capítulo se discutieron los resultados obtenidos con la correcta planificación de los tweets con el algoritmo bioinspirado seleccionado para tal efecto. Al aplicar las técnicas de extracción de datos para obtener las características relevantes del tweet que nos ayudaron a clasificar correctamente los patrones a detectar del suicidio en el contenido de este. Haciendo una validación con los algoritmos seleccionados para ayudarnos a detectar la mejor precisión de nuestro modelo con las técnicas de validación cruzada utilizadas. Logrando con esto optimizar los tiempos en el preprocesamiento de datos, concluyendo finalmente que con la estrategia bioinspirada propuesta en nuestra investigación se pueden lograr optimización de tiempos sustanciales que pueden ser cruciales para una correcta intervención de prevención del suicidio en los casos que se detecten en las redes sociales, no solamente de Twitter ya que esta metodología podría

ser utilizada en cualquier contexto de contenido ya sean páginas web, blogs y alguna otra red social como Facebook, Flickr, etc.

CONCLUSIONES GENERALES

La problemática del suicidio es un asunto de importancia global que requiere un enfoque multidisciplinario y la participación de todos los sectores de la sociedad para su prevención. La creación de un modelo automatizado para la detección de patrones relacionadas con el suicidio en plataformas como Twitter es una necesidad urgente. La metodología respaldada por técnicas de PNL y ML puede desempeñar un papel crucial en la identificación de este tipo de características entre los usuarios de las redes sociales, permitiendo una respuesta oportuna y potencialmente podría ayudar a prevenir el suicidio y con esto salvar vidas. Dadas las alarmantes tasas de suicidio en todo el mundo, existe una necesidad urgente de que la sociedad desempeñe un papel fundamental en la prevención del suicidio, ya que es crucial crear conciencia y el apoyo a las personas que enfrentan este problema.

Utilizando algoritmos bioinspirados basados en la evolución natural, ofrece una herramienta poderosa para abordar una variedad de problemas, incluido el análisis de datos relacionados con el suicidio. Sin embargo, la elección y el ajuste de algoritmos son fundamentales para el éxito, y es importante abordar estos enfoques con consideraciones éticas y responsabilidad, salvaguardando el anonimato de las personas. Con los resultados obtenidos en esta investigación se demostró que con técnicas de evolución natural es posible optimizar costos en los sistemas de cómputo en comparación con las arquitecturas secuenciales tradicionales, obteniendo una mejora de tiempo del 134% en el preprocesamiento de datos para la extracción de características relacionadas con el suicidio propuesta en la metodología, considerando que puede ser una alternativa viable que puede ayudar a la prevención del suicidio.

REFERENCIAS

- [1] W. H. Organization, "https://apps.who.int/iris/handle/10665/341728," 16 June 2021. [Online]. Available: <https://apps.who.int/iris/handle/10665/341728>. [Accessed 05 October 2022].
- [2] INEGI, "ESTADÍSTICAS A PROPÓSITO DEL DÍA MUNDIAL PARA LA PREVENCIÓN DEL SUICIDIO," 8 Septiembre 2021. [Online]. Available: https://www.inegi.org.mx/contenidos/saladeprensa/aproposito/2021/Suicidios2021_Nal.pdf. [Accessed Julio 2023].
- [3] A. i. p. p. d. suicidio, "https://www.iasp.info/es/index.php," 2020. [Online]. Available: <https://www.iasp.info/es/index.php>.
- [4] D. R. E. M. INEGI, "ESTADÍSTICAS DE DEFUNCIONES REGISTRADAS 2021," INEGI, 24 Enero 2021. [Online]. Available: https://www.inegi.org.mx/contenidos/programas/mortalidad/doc/defunciones_registradas_2021_nota_tecnica.pdf. [Accessed 3 Marzo 2023].
- [5] I. N. d. E. y. G. (INEGI), "ESTADÍSTICA DE DEFUNCIONES REGISTRADAS DE ENERO A JUNIO DE 2022 (PRELIMINAR)," june 2023. [Online]. Available: <https://www.inegi.org.mx/contenidos/saladeprensa/boletines/2023/DR/DR-Ene-jun2022.pdf>. [Accessed 01 March 2023].
- [6] T. M. a. A. M. I. L. López, "Fenómeno suicida: un acercamiento transdisciplinar,," in *Fenómeno suicida: un acercamiento transdisciplinar*, Editorial El Manual Moderno,, 2000.
- [7] N. N. G. P. D. M. D. D. L. & G. J. de Andrade, "Ethics and Artificial Intelligence: Suicide Prevention on Facebook,," *Philosophy & Technology*, vol. 31, no. 4, pp. 669-684, DOI: <https://doi.org/10.1007/s13347-018-0336-0>, 2018.
- [8] A. B.-H. M. E. Marouane Birjali, "Machine Learning and Semantic Sentiment Analysis based Algorithms for Suicide Sentiment Prediction in Social Networks,," *Procedia Computer Science*, vol. 113, pp. 65-72, DOI: <https://doi.org/10.1016/j.procs.2017.08.290>, 19 September 2017.
- [9] R. & S. Á. N. Urdiales Claros, "Síntomatología depresiva e ideación suicida como consecuencia de la pandemia por la COVID-19,," *Escritos de Psicología (Internet)*, vol. 14, no. 2, pp. 1138-2635, DOI: <https://dx.doi.org/10.24310/espiespsi.v14i2.12901>, 2021.
- [10] F. J. d. O. A. B. V. T. S. R. R. C. D. D. S. R. d. O. A. T. A. Diniz EJS, "Boamente: A Natural Language Processing-Based Digital Phenotyping Tool for Smart Monitoring of Suicidal Ideation,," *Healthcare (Basel)*, vol. 10, no. 4, pp. 698, DOI: <https://doi.org/10.3390%2Fhealthcare10040698>, 15 March 2022.
- [11] W. H. Organization, Comprehensive mental health action plan 2013–2030, World Health Organization, 2021.
- [12] O. Suicidio, "https://www.who.int/es/news-room/fact-sheets/detail/suicide," 17 junio 2021. [Online]. Available: <https://www.who.int/es/news-room/fact-sheets/detail/suicide>. [Accessed 15 Noviembre 2021].
- [13] D. R. E. M. INEGI, "CARACTERÍSTICAS DE LAS DEFUNCIONES REGISTRADAS EN MÉXICO DURANTE 2020, PRELIMINAR," INEGI, 29 Julio 2021. [Online]. Available: https://www.inegi.org.mx/contenidos/saladeprensa/boletines/2021/EstSociodemo/DefuncionesRegistradas2020_Pre_07.pdf. [Accessed 3 Noviembre 2021].
- [14] H. N. P. M. A. B. a. A. L. J. Pestian, "Suicide Note Classification Using Natural Language Processing: A Content Analysis,," *Biomedical Informatics Insights*, vol. 3, pp. 1-10, DOI: <https://doi.org/10.4137/BII.S4706>, Jan 2010.
- [15] J. Shapero, "The Language of Suicide Notes,," Ph. D. Thesis, 2011.
- [16] E. & P. d. A. A. Fonseca-Pedrero, "Evaluación de la conducta suicida en adolescentes: A propósito de la escala Paykel de suicidio,," *Papeles del psicólogo*, vol. 41, no. 2, pp. 106-115, DOI: <https://dx.doi.org/10.23923/pap.psicol2020.2928>, 2020.
- [17] M. R. Q. G. J. I. C. L. T. M. P. E. C. d. A. G. J. E. G. M. R. H. Sánchez-Loyo Luis Miguel, "Intento de Suicidio en Adolescentes Mexicanos: Perspectiva desde el Consenso Cultural,," vol. 4, no. 1, pp. 1446-1458, 2014.
- [18] K. P. O.-C. H. S. & M.-V. I. R. Valdés-García, "Aplicación de autopsias psicológicas: estudio de caso de dos suicidios en una familia,," *Psicumex*, vol. 11, no. 1, pp. 1-26, 2021.
- [19] S. G. M. K. N. B. S. M. A. M. M. S. K. B. Homan, "Linguistic features of suicidal thoughts and behaviors: A systematic review,," *Clinical psychology review*, vol. 95, p. 102161, 2022.
- [20] A. B.-H. M. E. Marouane Birjali, "A Method Proposed for Estimating Depressed Feeling Tendencies of Social Media Users Utilizing Their Data,," in *Proceedings of the 16th International Conference on Hybrid Intelligent Systems (HIS 2016)*, 2016.
- [21] X. H. Z. Z. L. J. Li LW, "An ecological study of social fragmentation, socioeconomic deprivation, and suicide in rural China: 2008–2010,," *SSM - Population Health*, vol. 2, pp. 365-372, ISSN 2352-8273, DOI: <https://doi.org/10.1016/j.ssmph.2016.05.007>, 2016.
- [22] B. A. Ahujaa R, "Mental Stress Detection in University Students using Machine Learning Algorithms,," *Procedia Computer Science*, vol. 152, pp. 349-353, 31 May 2019.
- [23] R. A. M. D. N. H. M. S. & C. H. Calvo, "Natural language processing in mental health applications using non-clinical texts,," *Natural Language Engineering*, vol. 23, no. 5, pp. 649-685, September 2017.
- [24] W. A. S. Ltd, "DIGITAL 2022: ANOTHER YEAR OF BUMPER GROWTH,," London, 2022.
- [25] L. H. X. B. Y. L. Tadesse MM, "Detection of Depression-Related Posts in Reddit Social Media Forum,," *IEEE Access*, vol. 7, pp. 44883 - 44893, DOI: <https://doi.org/10.1109/ACCESS.2019.2909180>, 4 Apr 2019.

- [26] J.-w. F. W.-I. Y. T. L. M. T. W. D. S. Z. Y. H. Manabu Torii, "Risk factor detection for heart disease by applying text analytics in electronic medical records," *Journal of biomedical informatics*, vol. 58, no. ISSN 1532-0464, pp. S164-S170, DOI: <https://doi.org/10.1016/j.jbi.2015.08.011>, Dec 2015.
- [27] T. O. B. L. M. Nguyen, "Using linguistic and topic analysis to classify sub-groups of online depression communities," *Science+Business Media*, vol. 76, no. April 2017, pp. 1573-7721, 21 December 2017.
- [28] J. Z. Y. L. J. Du, "Extracting psychiatric stressors for suicide from social media using deep learning," *BMC Medical Informatics and Decision Making*, vol. 18, no. 2, 2018.
- [29] M. R. A. M. E. L. P. J. B. A. L. C. H. C. Bridianne O'Dea, "The rate of reply and nature of responses to suicide-related posts on Twitter," *Internet Interventions*, vol. 13, pp. 105-107, ISSN 2214-7829, DOI: <https://doi.org/10.1016/j.invent.2018.07.004>, 2018.
- [30] É. Durkheim, Suicide, ISBN 0-203-99432-9 Master e-book ISBN, 1897.
- [31] T. F. D. H. S. R. J. A. Jacques Philip, "Relationship of Social Network to Protective Factors in Suicide and Alcohol Use Disorder Intervention for Rural Yup'ik Alaska Native Youth," *Psychosocial Intervention*, vol. 25, no. 1, pp. Pages 45-54, ISSN 1132-0559, DOI: <https://doi.org/10.1016/j.psi.2015.08.002>, 2016.
- [32] P. Liang and B. Dai, "Opinion mining on social media data," in *2013 IEEE 14th International Conference on Mobile Data Management 2013*, vol. 2, pp. 91-96, DOI: <https://doi.org/10.1109/MDM.2013.73>, 03 June 2013.
- [33] S. Brain, "Statistic Brain," 2014. [Online]. Available: <http://www.statisticbrain.com/twitter-statistics/>.
- [34] Z.-H. T. Y.-S. L. Y.-S. C. Chih-Hua T., "Mental Disorder Detection and Measurement Using Latent Dirichlet Allocation and SentiWordNet," *2015 IEEE International Conference on Systems, Man, and Cybernetics, Kowloon, 2015*, pp. 1215-1220, DOI: <https://doi.org/10.1109/SMC.2015.217>, 09-12 October 2015.
- [35] D. P. Phillips, "The Influence of Suggestion on Suicide: Substantive and Theoretical Implications of the Werther Effect," *American Sociological Review*, vol. 39, no. 3, pp. 340-354, DOI: <https://doi.org/10.2307/2094294>, Jun 1974.
- [36] F. B. E. du Roscoët, "Efficient interventions on suicide prevention: A literature review," *Revue d'Épidémiologie et de Santé Publique*, vol. 61, no. 4, pp. 363-374, August 2013.
- [37] S. A. A. J. S. B. C. L. & T. J. Renjith, "An ensemble deep learning technique for detecting suicidal ideation from posts in social media platforms," *Journal of King Saud University-Computer and Information Sciences*, 2021.
- [38] S. E. A. & B. P. Ghosh, "A multitask framework to detect depression, sentiment and multi-label emotion from suicide notes," *Cognitive Computation*, vol. 14, no. 1, pp. 110-129, 2022.
- [39] P. Moron, El Suicidio, 11 ed., L. A. SA, Ed., Mexico: Publicaciones Cruz O., SA, 1987.
- [40] G. H. Amador Rivera, "Suicidio: consideraciones históricas," *Revista Médica La Paz*, vol. 21, no. 2, pp. 91-98, 2015.
- [41] G. H. A. Rivera, "SUICIDIO: CONSIDERACIONES HISTÓRICAS," *Revista Médica La Paz*, vol. 21, no. 2, pp. 91-98., 20158.
- [42] A. M. Romero Valle, "El suicidio a finales del siglo XIX (1899). Visiones predominantes en la prensa," Universidad Nacional Autónoma de México, 2001. [Online]. Available: <http://132.248.9.195/pd2001/288327/288327.pdf>.
- [43] A. F. Palacio, "La comprensión clásica del suicidio. De Émile Durkheim a nuestros días," *Affectio societatis*, vol. 7, no. 12, p. 6, 2010.
- [44] INEGI, "COMUNICADO DE PRENSA NÚM. 503/22 8 DE SEPTIEMBRE DE 2022," INEGI, Mexico, 2022.
- [45] INEGI, "Estadística de Intentos de Suicidio y Suicidio," INEGI, MEXICO, 1994.
- [46] INEGI, "Estadísticas de Intentos de Suicidio y Suicidios, Cuaderno Núm. 1," Instituto Nacional de Estadística, Edificio Sede, Aguascalientes, Ags, 1995.
- [47] INEGI, "Estadísticas de Intentos de Suicidio y Suicidios, Cuaderno Núm. 2," Instituto Nacional de Estadística. Geografía e Informática, Edificio Sede, Aguascalientes, Ags, 1997.
- [48] INEGI, "Estadísticas de Intentos de Suicidio y Suicidios, Cuaderno Núm. 10," Instituto Nacional de Estadística, Geografía e Informática. Edificio Sede, Aguascalientes, Ags., 2004.
- [49] INEGI, "Estadística de suicidios de los Estados Unidos Mexicanos 2006," Instituto Nacional de Estadística, Geografía e Informática, Aguascalientes, Ags., 2006.
- [50] INEGI, "Defunciones registradas por suicidio por entidad federativa y causa según sexo, serie anual de 2010 a 2021," INEGI, Aguascalientes, AGS, 2023.
- [51] INEGI, "Suicidios según entidad federativa 2010-2021," INEGI, Aguascalientes, Ags, 2022.
- [52] INEGI, "ESTADÍSTICAS A PROPÓSITO DEL DÍA MUNDIAL PARA LA PREVENCIÓN DEL SUICIDIO," INEGI, Aguascalientes, Ags, 2021.
- [53] O. P. d. I. Salud, "Vivir la vida. Guía de aplicación para la prevención del suicidio en los países," *World Health Organization*, 2021.
- [54] W. P. W. McCulloch, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, vol. 5, p. 115-133, 1943.
- [55] D. O. Hebb, The organization of behavior: A neuropsychological theory, New York: McGill University, 1949.
- [56] A. L. Samuel, "Some Studies in Machine Learning Using the Game of Checkers," *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210-229, 1959.
- [57] F. Rosenblatt, "The perceptron, a perceiving and recognizing automaton," *Cornell Aeronautical*, 1957.

- [58] A. G. & L. V. G. Ivakhnenko, Cybernetic predicting devices, 1966, pp. 1-214.
- [59] T. C. a. P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21-27, January 1967.
- [60] K. Fukushima, "Neocognitron: A hierarchical neural network capable of visual pattern recognition.," *Neural networks*, vol. 2, no. 1, pp. 119-130, 1988.
- [61] C. V. V. Cortes, "Support-vector networks," *Mach Learn*, vol. 20, p. 273-297, 1995.
- [62] R. Schapire, "The strength of weak learnability," *Mach Learn*, pp. 197-227, June 1990.
- [63] D. H. G. & W. R. Rumelhart, "Learning Representations by Backpropagation Error," *Nature*, pp. 533-536, 31 July 1986.
- [64] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5-32, October 2001.
- [65] I. P.-A. J. M. M. X. B. W.-F. D. O. S. ... & B. Y. Goodfellow, "Generative adversarial nets," in *Advances in neural information processing systems*, 27..
- [66] P. a. t. P. Jeffrey M. Stanton (2001) Galton, "A Brief History of Linear Regression for Statistics Instructors," *Journal of Statistics Education*, pp. 9,3, 2017.
- [67] J. P. J. & T. H. Han, Data mining: concepts and techniques, 500 Sansome Street, Suite 400, San Francisco, CA 94111: Morgan kaufmann, 2022.
- [68] J. B. M.D, "Application of the Logistic Function to Bio-Assay," *Journal of the American Statistical Association*, p. 39:227, 1944.
- [69] J. C. Stoltzfus, "Logistic regression: a brief primer," *Academic emergency medicine*, vol. 18, no. 10, pp. 1099-1104, 2011.
- [70] E. J. C. Suárez, "Tutorial sobre máquinas de vectores soporte (sVM)," *Tutorial sobre Máquinas de Vectores Soporte (SVM)*, vol. 1, pp. 1-12, 2014.
- [71] M. & T. P. N. Steinbach, kNN: k-nearest neighbors. The top ten algorithms in data mining, Boca Raton, FL 33487-2742: Chapman & Hall/CRC, Taylor & Francis Group, 2009, pp. 151-162.
- [72] J. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825--2830, 2011.
- [73] H. W. Youguo Li, "A Clustering Method Based on K-Means Algorithm," *Physics Procedia*, vol. 25, pp. 1104-1109, 2012.
- [74] R. Sibson, "SLINK: An optimally efficient algorithm for the single-link cluster method," *The Computer Journal*, vol. 16, no. 1, pp. 30-34, 1973.
- [75] H. & W. L. J. Abdi, "Principal component analysis.," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433-459, 2010.
- [76] L. Van Der Maaten, "Accelerating t-SNE using tree-based algorithms," *The journal of machine learning research*, vol. 15, no. 1, pp. 3221-3245, 2014.
- [77] F. V. G. G. A. M. V. T. B. G. O. o. Pedregosa, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, p. 2825-2830, 2011.
- [78] D. M. N. A. Y. & J. M. I. Blei, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, pp. 993-1022, Jan 2003.
- [79] C. J. C. H. Watkins, Learning from delayed rewards., Doctoral dissertation ed., Cambridge, United Kingdom: University of Cambridge, 1989.
- [80] J. J. H. A. P. N. H. T. E. Y. T. D. S. D. W. Timothy P. Lillicrap, "Continuous control with deep reinforcement learning," *arXiv*, vol. 6, Sep 2015.
- [81] J. W. F. D. P. R. A. & K. O. Schulman, "Proximal Policy Optimization Algorithms," *arXiv*, 20 July 2017.
- [82] H. C. P. H. W. P. M. & Y. Szu, "Collective and distributive swarm intelligence: evolutionary biological survey," in *International Congress Series*, 2004.
- [83] A. S. & B. E. Radhamani, "LOW COMPLEXITY CONSTRAINTS FOR ENERGY AND PERFORMANCE MANAGEMENT OF HETEROGENEOUS MULTICORE PROCESSORS USING DYNAMIC OPTIMIZATION," *Journal of Computer Science*, vol. 10, no. 8, pp. 1508-1516, 2014.
- [84] G. S. J. B. X. & M. M. Reynoso-Meza, "Algoritmos Evolutivos y su empleo en el ajuste de controladores del tipo PID: Estado Actual y Perspectivas," *Revista Iberoamericana de Automática e Informática Industrial RIAI*, vol. 10, no. 3, pp. 251-268, 2013.
- [85] N. & S. S. Chitaly, "Performance analysis of GA, PSO and JA for determining the optimal parameters in friction drilling process," *Engineering Science and Technology, an International Journal*, p. 101246, 2022.
- [86] M. K. K. M. R. & T. C. R. Patel, "A hybrid ACO/PSO based algorithm for QoS multicast routing problem," *Ain Shams Engineering Journal*, vol. 5, no. 1, pp. 113-120, 2014.
- [87] J. W. W. L. S. D. Q. & C. L. Sun, "Scheduling optimization for UAV communication coverage using virtual force-based PSO model.," *Digital Communications and Networks*, 2023.
- [88] S. A. A. A. D. & S. M. A. Alsaidy, "Heuristic initialization of PSO task scheduling algorithm in cloud computing," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, pp. 2370-2382, 2022.
- [89] R. a. J. K. Eberhart, "A new optimizer using particle swarm theory.," in *MHS'95. Proceedings of the sixth international symposium on micro machine and human science*, 1995.

- [90] Y. S. a. R. Eberhart, "A modified particle swarm optimizer," *IEEE world congress on computational intelligence*, pp. 69-73, May 1998.
- [91] M. B. M. & S. T. Dorigo, "Ant colony optimization," *IEEE computational intelligence magazine*, vol. 1, no. 4, pp. 28-39, 2006.
- [92] H. John, *Adaptation in natural and artificial systems.*, University of Michigan Press.: Ann Arbor., 1975.
- [93] M. & K. M. H. Nikravan, "A genetic algorithm for process scheduling in distributed operating systems considering load balancing," in *n Proc. of 21st European Conference on Modelling and Simulation*, Europe, 2007.
- [94] E. G. J. J. D. N. M. A. L. & M. M. Azketa, "Algoritmo genético permutacional para el despliegue y la planificación de sistemas de tiempo real distribuidos," *Revista Iberoamericana de Automática e Informática industrial*, vol. 10, no. 3, pp. 344-355, 2013.
- [95] S. & C. N. K. Sandhya, "Dynamic Load Balancing Based on Genetic Algorithm," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 11, pp. 176-179, 2019.
- [96] P. & S. S. M. Shah, "Load Balancing in Distributed System Using Genetic Algorithm," vol. 1, pp. 139-142, 2011.
- [97] S. S. a. J. Palmucci, "The application of genetic algorithms to resource scheduling," in *In Proceedings of the 4th International Conference on Genetic Algorithms*, ICGA, 1991.
- [98] I. M. S. D. & H. J. R. Oliver, "Study of permutation crossover operators on the traveling salesman problem," *Proceedings of the second International Conference on Genetic Algorithms*, pp. 224-230, 31 July 1987.
- [99] H. Sueki, "The association of suicide-related Twitter use with suicidal behaviour: a cross-sectional study of young internet users in Japan," *Journal of Affective Disorders*, vol. 170, pp. 155-160, 2015.
- [100] S. W. P. J. B. A. L. C. C. P. H. C. Bridianne O'Dea, "Detecting suicidality on Twitter," *Internet Interventions*, vol. 2, no. 2, pp. 183-188, 2015.
- [101] L. M. James G. Phillips, "Suicide baiting in the internet era," *Computers in Human Behavior*, vol. 92, pp. 29-36, 2019.
- [102] A. A. C. E. F. Cheryl A King, "Emerging trends in adolescent suicide prevention research," *Current Opinion in Psychology*, vol. 22, pp. 89-94, 2018.
- [103] Y. K. S. Y. Lee, "Twitter as a place where people meet to make suicide pacts," *Public Health*, vol. 159, p. 21.26, 2018.
- [104] T. M. M. U. Robert A. Fahey, "Tracking the Werther Effect on social media: Emotional responses to prominent suicide deaths on twitter and subsequent increases in suicide," *Social Science & Medicine*, vol. 219, pp. 19-29, ISSN 0277-9536, <https://doi.org/10.1016/j.socscimed.2018.10.004>, December 2018.
- [105] B. T. D. G. Thomas Niederkrotenthaler, "Celebrity suicide on Twitter: Activity, content and network analysis related to the death of Swedish DJ Tim Bergling alias Avicii," *Journal of Affective Disorders*, vol. 245, pp. 848-855, 2019.
- [106] G. C. R. A. A. H. J. S. Pete Burnap, "Multi-class machine classification of suicide-related communication on Twitter," *Online Social Networks and Media*, vol. 2, pp. 32-44, 2017.
- [107] V. H. Bart Desmet, "Online suicide prevention through optimised text classification," *Information Sciences*, vol. 439-440, pp. Pages 61-78, May 2018 2018.
- [108] A. C. D. E. J. T. S. & T. A. S. de Oliveira, "How can machine learning identify suicidal ideation from user's texts? Towards the explanation of the Boamente system," *Procedia Computer Science*, vol. 206, pp. 141-150, 2022.
- [109] M. M. R. P. & R. I. A. Casado Blanco, "Importancia de las cartas suicidas en la investigación forense," *Cuadernos de Medicina Forense*, vol. 18, no. 3, pp. 3-4, 2012.
- [110] S. Z. S. J. M. G. & N.-C. B. Cremades, "Diseño, compilación y anotación de un corpus para la detección de mensajes suicidas en redes sociales," *Procesamiento del Lenguaje Natural*, vol. 59, pp. 56-72, 2017.
- [111] S. S. A. S. P. a. B. A. Cheeda, "Automated trading of cryptocurrency using twitter sentimental analysis," in *J. Comput. Sci. Eng*, 2018.
- [112] A. B. X. I. V. O. R. a. R. J. P. Agarwal, "Sentiment analysis of twitter data," in *In Proceedings of the Workshop on Language in Social Media*, New York, USA, 2011.
- [113] B. McDonalds, "University of Notre Dame," University of Notre Dame, 10 April 2020. [Online]. Available: <https://www3.nd.edu/~mcdonald/>.
- [114] R. W. A. S. J. M. G. & S. H. A. M. Caicedo, "Assessment of supervised classifiers for the task of detecting messages with suicidal ideation," *Heliyon*, vol. 8, p. 6, 2020.
- [115] T. C. K. C. G. & D. J. Mikolov, "Efficient estimation of word representations in vector space," *Computation and Language*, vol. 1381, p. 1301, 2013.
- [116] S. & C. A. Arlot, "A survey of cross-validation procedures for model selection," *Statistics surveys*, vol. 4, pp. 40-79, 2010.
- [117] C. F. B. a. A. P. Maaoui, "Automatic human stress detection based on webcam photoplethysmographic signals," *Journal of Mechanics in Medicine and Biology*, vol. 16, no. 4, 2016.
- [118] Y.-H. T. A. Y. Zomaya, "Observations on using genetic algorithms for dynamic load-balancing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 12, no. 9, pp. 899-911, 2001.
- [119] T. a. C. A. M. Basu, "A feature selection method for improved document classification," *Advanced Data Mining and Applications*, vol. 7713, pp. 296-305, 2012.
- [120] V. H. Bart Desmet, "Emotion detection in suicide notes," *Expert Systems with Applications*, vol. 40, no. 16, pp. 6351-6358, 2013.