



UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO
ESCUELA DE POSGRADO

MAESTRÍA EN ECONOMÍA MENCIÓN PROYECTOS DE INVERSIÓN

TESIS

**ALGORITMO GENÉTICO Y RED NEURONAL ARTIFICIAL
PARA LA OPTIMIZACIÓN DE PORTAFOLIOS DE INVERSIÓN**

**PARA OPTAR AL GRADO ACADÉMICO DE MAESTRO EN
ECONOMÍA MENCIÓN PROYECTOS DE INVERSIÓN**

AUTOR

**Br. GUSTAVO ALEXIS FLORES
FERNANDEZ**

ASESOR:

Mg. ERICK CHUQUITAPA ROJAS

CÓDIGO ORCID:

0000-0003-2635-1389

CUSCO-PERÚ

2024

INFORME DE ORIGINALIDAD

(Aprobado por Resolución Nro.CU-303-2020-UNSAAC)

El que suscribe, **Asesor** del trabajo de investigación/tesis de maestría titulada: "ALGORITMO GENÉTICO Y RED NEURONAL ARTIFICIAL PARA LA OPTIMIZACIÓN DE PORTAFOLIOS DE INVERSIÓN"; presentado por el Bach. GUSTAVO ALEXIS FLORES FERNANDEZ con DNI Nro 47170879 para optar el título profesional/grado académico de MAESTRO EN ECONOMÍA, MENCIÓN PROYECTOS DE INVERSIÓN. Informo que el trabajo de investigación ha sido sometido a revisión por 02 veces, mediante el Software Antiplagio, conforme al Art. 6° del **Reglamento para Uso de Sistema Antiplagio de la UNSAAC** y de la evaluación de originalidad se tiene un porcentaje de ocho por ciento (8%)

Evaluación y acciones del reporte de coincidencia para trabajos de investigación conducentes a grado académico otítulo profesional, tesis

Porcentaje	Evaluación y Acciones	Marque con una (X)
Del 1 al 10%	No se considera plagio.	X
Del 11 al 30 %	Devolver al usuario para las correcciones.	
Mayor a 31%	El responsable de la revisión del documento emite un informe al inmediato jerárquico, quien a su vez eleva el informe a la autoridad académica para que tome las acciones correspondientes. Sin perjuicio de las sanciones administrativas que correspondan de acuerdo a Ley.	

Por tanto, en mi condición de asesor, firmo el presente informe en señal de conformidad y **adjunto** la primera página del reporte del Sistema Antiplagio.

Cusco, 26 de agosto de 2024



Firma

ERICK CHUQUITAPA ROJAS

Nro. de DNI 40859559

ORCID del Asesor: <https://orcid.org/0000-0003-2635-1389>

Se adjunta:

1. Reporte generado por el Sistema Antiplagio.
2. Enlace del Reporte Generado por el Sistema Antiplagio: **oid:27259:375381609**

NOMBRE DEL TRABAJO

**ALGORITMO GENÉTICO Y RED NEURON
AL ARTIFICIAL PARA LA OPTIMIZACIÓN
DE PORTAFOLIOS DE INVERSIÓN**

AUTOR

Gustavo Alexis Flores Fernandez

RECUESTO DE PALABRAS

26655 Words

RECUESTO DE CARACTERES

139545 Characters

RECUESTO DE PÁGINAS

99 Pages

TAMAÑO DEL ARCHIVO

2.8MB

FECHA DE ENTREGA

Aug 23, 2024 11:44 AM GMT-5

FECHA DEL INFORME

Aug 23, 2024 11:46 AM GMT-5**● 8% de similitud general**

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para cada base de datos.

- 8% Base de datos de Internet
- Base de datos de Crossref
- 1% Base de datos de publicaciones
- Base de datos de contenido publicado de Crossref

● Excluir del Reporte de Similitud

- Base de datos de trabajos entregados
- Material bibliográfico
- Material citado
- Coincidencia baja (menos de 10 palabras)
- Material citado



DEDICATORIA

En primer lugar, a Dios, por su inmensa bondad y guía en todo acto de perseverancia y esfuerzo hacia mis metas y objetivos.

A mis padres que siempre me han acompañado y me han dado ánimos en cada paso profesional y personal que he realizado.

A Manuel Flores, Hipólito Fernández y Lucas Flores, que además de ser familiares fueron mentores.

AGRADECIMIENTO

Un agradecimiento a mis asesores, Mgtr. Erick Chuquitapa Rojas y Dr. Miguel Jiménez Carrión por su apoyo durante toda la investigación

Un agradecimiento a los docentes de la maestría que con el conocimiento impartido fue posible el desarrollo de esta investigación

INDICE

RESUMEN	ix
ABSTRACT	x
INTRODUCCIÓN.....	x
CAPITULO I: PLANTEAMIENTO DEL PROBLEMA.....	1
1.1. Situación problemática	1
1.2. Formulación del problema.....	3
1.3. Justificación de la investigación	4
1.4. Objetivos de la investigación.....	4
b. Objetivos Específicos	4
1.5. Delimitación de la investigación	5
a. Delimitación temporal	5
b. Delimitación espacial	5
c. Delimitación económica	5
CAPITULO II: MARCO TEÓRICO CONCEPTUAL	6
2.1. Bases teóricas	6
2.1.1. Teoría de carteras de inversión	6
2.1.2. Complejidad computacional	13
2.1.3. Metaheurística y la optimización combinatoria.....	14
2.1.4. Algoritmos Genéticos	15
2.1.5. Redes neuronales artificiales	23
2.2. Marco Conceptual.....	33
2.3. Antecedentes empíricos de la investigación.....	35
2.4 Hipótesis	39
2.5 Identificación de variables e indicadores.....	39
2.6. Operacionalización de variables	41
CAPITULO III: METODOLOGÍA.....	42
3.1. Ámbito de estudio.....	42
3.2. Tipo y nivel de investigación.....	42
3.3. Unidad de análisis.....	42
3.4. Población de estudio.....	42
3.5. Tamaño de muestra.....	42
3.6. Técnicas de selección de muestra.....	43

3.7. Técnicas de recolección de información	43
3.8. Técnicas de análisis e interpretación de la información	43
3.9. Técnicas para demostrar la verdad o falsedad de las hipótesis planteadas.....	43
CAPITULO IV: RESULTADOS Y DISCUSIÓN	43
4.1. Resultados.....	44
4.1.1. Red Neuronal artificial para predecir la tendencia de los activos de capital	44
4.1.2. Modelamiento de los elementos del algoritmo genético	63
4.1.3. Evaluación del algoritmo genético	70
4.2. Discusión	77
CAPITULO V: CONCLUSIONES Y RECOMENDACIONES	79
5.1. Conclusiones.....	79
5.2. Recomendaciones	80
BIBLIOGRAFIA	81

INDICE DE TABLAS

Tabla 1 <i>Representación Poncentual de empresas según datos faltantes</i>	46
Tabla 2 <i>Datos faltantes del registro de acciones de las empresas</i>	49
Tabla 3 <i>Variaciones Poncentuales después de la imputación polinómica</i>	50
Tabla 4 <i>Variaciones Poncentuales después de la imputación por medianas</i>	51
Tabla 5 <i>Imputación por el método KNN</i>	53
Tabla 6 <i>Comparación de coeficientes de determinación de los métodos de imputación de datos</i>	55
Tabla 7 <i>Pseudocodigo de la red neuronal artificial</i>	57
Tabla 8 <i>Comparación de modelos de red LSTM</i>	58
Tabla 9 <i>Comparación de las métricas de error de los modelos</i>	59
Tabla 10 <i>Comportamiento de las variables lambda en cuanto a su correlación y variacion asociados al fitness</i>	68
Tabla 11 <i>Proporcion asignada de lambda's a cada factor del fitness</i>	68
Tabla 12 <i>Comparación de funciones objetivo segun la configuración del algoritmo genetico</i>	71
Tabla 13 <i>Repeticiones para cada configuración del algoritmo genetico</i>	71
Tabla 14 <i>Analisis de varianza</i>	72
Tabla 15 <i>Prueba de Tukey de diferencias de medias</i>	72
Tabla 16 <i>Prueba de Duncan</i>	73
Tabla 17 <i>Pseudocodigo del algoritmo genetico</i>	74
Tabla 18 <i>Resultados de Poncentajes a invettir en cada activo perteneciente al portafolio</i>	76

INDICE DE FIGURAS

Figura 1 <i>Fronteras de Pareto</i>	8
Figura 2 <i>Optimo en el modelo CAPM</i>	9
Figura 3 <i>Conjuntos de complejidad computacional</i>	14
Figura 4 <i>Conceptualización de población e individuos en algoritmos genéticos</i>	16
Figura 5 <i>Proceso de cruzamiento en algoritmos genéticos</i>	19
Figura 6 <i>Representación de cruce de n puntos</i>	19
Figura 7 <i>Representación de cruce uniforme</i>	19
Figura 8 <i>Representación de cruce por emparejamiento parcial</i>	20
Figura 9 <i>Operación de cruzamiento CxBlend</i>	21
Figura 10 <i>Representación de cruce en orden</i>	21
Figura 11 <i>Representación de cruce por ciclos</i>	22
Figura 12 <i>Operación de mutación</i>	22
Figura 13 <i>Mutación Gaussiana</i>	23
Figura 14 <i>Neurona artificial</i>	23
Figura 15 <i>Estructura de red neuronal artificial</i>	25
Figura 16 <i>Función escalón</i>	26
Figura 17 <i>Función lineal</i>	26
Figura 18 <i>Función Sigmoide</i>	27
Figura 19 <i>Función de Gauss</i>	27
Figura 20 <i>Esquema de aprendizaje supervisado</i>	28
Figura 21 <i>Esquema de aprendizaje no supervisado</i>	29
Figura 22 <i>Ejemplo de conexiones recurrentes</i>	30
Figura 23 <i>Celda LSTM</i>	31
Figura 24 <i>Representación gráfica de las acciones de las empresas antes del tratamiento de outliers y datos faltantes</i>	48
Figura 25 <i>Representación de comportamiento de los datos imputados</i>	54
Figura 26 <i>Representación de RMSE de la empresa AIHC1</i>	59
Figura 27 <i>Representación de RMSE de la empresa GBVLAC1</i>	60
Figura 28 <i>Comparación de datos de validación de la empresa GBVLAC1 con las predicciones del modelo</i>	60
Figura 29 <i>Residuos generados en el set de validación</i>	61
Figura 30 <i>Representación del RMSE para cada empresa evaluada</i>	61
Figura 31 <i>Histograma de errores</i>	62
Figura 32 <i>Representación cromosomica de los individuos</i>	64
Figura 33 <i>Representación de la función asociada a la prima de mercado</i>	66
Figura 34 <i>Correlación de las variables lambda con la maximización del fitness</i>	68
Figura 35 <i>Estabilidad del algoritmo en relacion a su fitness</i>	74
Figura 36 <i>Estabilidad del algoritmo en relacion a su rentabilidad y volatilidad</i>	75
Figura 37 <i>Frontera de pareto rendimiento vs volatilidad</i>	75

RESUMEN

La presente investigación tuvo como objetivo general construir un algoritmo genético y red neuronal artificial para optimizar portafolios de inversión, para tal efecto, se evaluaron 255 empresas que realizan sus actividades dentro del ámbito nacional peruano y cotizan sus acciones en la Bolsa de Valores de Lima. En esta investigación se usó una metodología de tipo aplicada, de diseño experimental y de corte transversal en donde se obtuvieron como resultados una raíz del error cuadrático medio de 6.33% y un error absoluto medio de 5.07% y una precisión de 92.35% relacionada con la red neuronal artificial para predecir la tendencia positiva de las acciones que serán entradas para el algoritmo genético, así también se lograron definir los hiperparámetros del algoritmo genético como la probabilidad de cruce del 0.8 y la probabilidad de mutación del 0.09, de la misma forma se modeló la función de calidad que consideró 5 factores, finalmente se encontró como mejor configuración del algoritmo genético la arquitectura que tuvo como fitness un valor de 0.772482, que se traduce en una rentabilidad de 1.00058% que se traduce en un 12.00696 % anual y volatilidad de 0.00612188%. Se concluye que el algoritmo genético optimiza portafolios de inversión al encontrarse una rentabilidad mayor y una volatilidad menor comparada con otros métodos.

Palabras clave: metaheurísticas, algoritmos genéticos, red neuronal artificial, optimización, portafolios de inversión.

ABSTRACT

The general objective of this research was to develop a genetic algorithm and an artificial neural network to optimize investment portfolios. To this end, 255 companies operating within the Peruvian national market and listed on the Lima Stock Exchange were evaluated. This research employed an applied methodology with an experimental design and cross-sectional approach. The results included a root mean square error of 6.33%, a mean absolute error of 5.07%, and an accuracy of 92.35% for the artificial neural network in predicting the positive trend of stocks that would serve as input for the genetic algorithm. Additionally, the hyperparameters of the genetic algorithm were defined, with a crossover probability of 0.8 and a mutation probability of 0.09. The fitness function was modeled considering five factors. Finally, the best configuration of the genetic algorithm achieved a fitness value of 0.772482, translating to a profitability of 1.00058%, which corresponds to an annual return of 12.00696% and a volatility of 0.00612188%. It was concluded that the genetic algorithm optimizes investment portfolios by achieving higher profitability and lower volatility compared to other methods.

Keywords: metaheuristics, genetic algorithms, artificial neural networks, optimization, investment portfolios.

INTRODUCCIÓN

La optimización de recursos siempre ha sido un tema relevante y aún más representativo en el ámbito económico, siendo así que los proyectistas enfrentan ciertas situaciones en donde deben buscar métodos para mejorar la rentabilidad de las empresas representadas en los estados financieros que reflejen de una manera fiel a la realidad, el aprovechamiento de exceso de liquidez, sin embargo esta tarea no es sencilla debido a que las fuentes de inversión que generen rentabilidades atractivas son pocas, por lo que una opción viable es el uso de carteras o portafolios de inversión en el mercado de valores o activos de capital, en donde a partir de que los inversionistas asuman un riesgo, pueden obtener una rentabilidad aceptable, sin embargo conformar estos portafolios de inversión no es una tarea sencilla, ya que se debe escoger cuidadosamente las proporciones de capital que se asignaran a los activos para tratar de minimizar la volatilidad y maximizar la rentabilidad. En este escenario se plantea esta investigación, considerando que las múltiples combinaciones que existen de porcentajes a asignar convierten este problema en uno de tipo no polinomial completo, por lo que se hace relevante el uso de herramientas de inteligencia artificial como los son las redes neuronales artificiales y los algoritmos genéticos para poder obtener resultados óptimos favorables y que puedan satisfacer las condiciones de optimizar un portafolio tanto en rentabilidad como en volatilidad.

Para el desarrollo de esta investigación se hizo uso de los datos públicos de los rendimientos de los activos de valores que cotizan en la bolsa de valores de Lima, estos datos pasaron por un análisis exploratorio para mejorar su calidad y luego ser procesados por la red neuronal cuya funcionalidad fue ser un primer filtro para seleccionar los activos de capital que tuvieron un pronóstico favorable, garantizándose que el portafolio tengan rendimientos positivos, y luego pasar al procesamiento del algoritmo genético cuya función fue obtener portafolios óptimos determinándose los porcentajes de inversión a cada activo de capital que conforma el portafolio.

CAPITULO I: PLANTEAMIENTO DEL PROBLEMA

1.1. Situación problemática

A nivel mundial se puede evidenciar una difícil situación económica en las unidades empresariales o productivas, esto debido a la incertidumbre generada por las condiciones en las que se desarrolla el comportamiento de mercado en las economías externas e internas, el problema de la inflación y otros factores; que hacen que las organizaciones busquen gestionar de forma más efectiva sus proyectos y mejorar las inversiones, tal como se referencia de Maholi Solin et al. (2019), la inversión tiene un rol fundamental relacionado a la mejora de la economía de un país, ya que una mayor inversión se traduce en mayor valor obtenido, y por lo tanto, mayor será la capacidad del país para desarrollarse más rápido, esto se refleja a nivel internacional, en el caso de países como la Federación Rusa, en donde sus inversiones en el 2018 aumentaron en 4.3% para mitigar este tipo de problemas, haciendo mención de datos obtenidos en el estudio de Chkalova et al. (2020). En este sentido, las unidades productivas que dinamizan las economías locales y nacionales, tienen como principal necesidad, la planificación en sus inversiones, para adecuarse mejor a su demanda; haciendo referencia a lo dicho por Baca (2016) en donde menciona que esto implica generalmente evaluar en un horizonte de planeamiento, el futuro escalamiento de infraestructura u operaciones, mediante la evaluación de proyectos de inversión. Muchas veces la evaluación de proyectos de inversión no contempla dentro de sus estados financieros proyectados, escenarios realistas y detalles específicos propios de las operaciones de las unidades productoras, como son la adquisición de activos fijos en los períodos transitorios de escalabilidad empresarial, inversiones no tangibles, etc. En referencia a lo que mencionan Hart y Zingales (2011), la necesidad de invertir en activos, que muchas veces no son contemplados en los estados financieros, evidencia la falta de herramientas para mejorar la eficiencia de las inversiones (p.4.), evidenciando el desaprovechamiento del exceso de liquidez en la unidad productora, que conllevaría a que los flujos netos acumulados reflejen de forma más adecuada el dinamismo empresarial, y no caer en rendimientos negativos sin haber intentado mejorar la eficiencia en inversiones externas.

El proyectista que se enfrente a un exceso de liquidez o riqueza en sus flujos proyectados debe tomar una decisión acertada para mejorar la eficiencia del proyecto, y ello conlleva decidir, si toma los costos de oportunidad, entre los cuales se puede contemplar el mantener el dinero, evitando o asumiendo los cambios inflacionarios, o hacer una inversión libre de riesgo invirtiendo en una entidad financiera, o en caso contrario; citando a lo que

indica Ames (2012), invirtiendo en el mercado de capitales, lo cual es una opción viable para los jefes de proyectos que asumen un riesgo, y en donde encontrarán un refugio de valor con esto se puede tener como consecuencia un rendimiento mayor a la inversión en una entidad financiera, y que para el caso de la presente investigación la oportunidad evaluada fue la Bolsa de valores de Lima, considerando que es dentro del ámbito nacional en donde se desarrolla la investigación y es un caso específico en donde se puede encontrar un refugio de valor de manera más accesible. Sin embargo se encuentran problemas como los descritos por autores como Cheong et al. (2017) que afirman que no existe una política de gestión activa de carteras de inversión de manera efectiva por parte de los inversores que tengan excedentes de liquidez y esto también se ve reflejado en el ámbito nacional en donde no se hace evidente el uso de este tipo de herramientas, lo cual evidencia la necesidad de uso de estrategias innovadoras como el uso de algoritmos especializados. Por lo tanto frente al reto que representa determinar los elementos de una cartera de inversión de forma óptima, el proyectista debe tomar una decisión al escoger la mejor opción para diversificar el portafolio y esa decisión es un problema altamente combinatorio, en referencia a lo manifestado por Didier et al. (2021) cuando afirman que la opción de generar rendimiento en el mercado de capitales, considerando que existe un riesgo, es de vital importancia escoger de manera eficiente los activos, implicando un rendimiento mayor en la inversión, y minimizar el riesgo asumible relacionado con la volatilidad de los mismos activos (p.2.), por lo que autores como Granada y Santa (2016) indican que este tipo de problemas ha motivado el crecimiento acelerado de metaheurísticas como son las redes neuronales y algoritmos genéticos; donde miles de problemas del mundo real pueden ser definidos como problemas abstractos de optimización combinatoria. En específico al considerar que cada activo de capital se le debe asignar un porcentaje de inversión del excedente de los flujos proyectados, se tendría un proceso altamente combinatorio de opciones, por ejemplo si se evaluaran 5 activos, se tendrían 10,000,000,000 combinaciones, siendo las representaciones valores porcentuales, además si cada combinación se evaluará en una función objetivo con una demora de tiempo de 1 segundo, se tendría un tiempo exponencial en poder evaluar todas las combinaciones para 10 o 20 activos. Por lo tanto, este problema con dos objetivos contrapuestos; por ser de maximización en el caso del rendimiento, y de minimización en el caso del riesgo, no puede ser resuelto en todas sus combinaciones en un tiempo razonable mediante un algoritmo determinista, y considerando su naturaleza combinatoria se categoriza finalmente como un problema “no polinomial” esto en referencia a lo dicho por Gutiérrez et al. (2020). Si bien es cierto que en la evaluación de portafolios de inversión se tiende a realizar la evaluación

de la cartera con una cantidad determinada de combinaciones para representar la frontera de Pareto y utilizar el índice de Sharpe para encontrar el óptimo, no existe una solución determinista que contemple y evalúe de forma iterativa una cantidad significativa de soluciones eficientes y mucho menos que contemple la optimización de los dos objetivos al mismo tiempo para encontrar soluciones factibles. Por lo que se hace de vital importancia el uso de herramientas de inteligencia artificial como son las redes neuronales artificiales y metaheurísticas como los algoritmos genéticos, tal es el caso de autores como Das et al. (2023) y Li et al. (2023), que a partir de sus hallazgos de puede concluir la utilidad fundamental en el campo de las finanzas de estas tecnologías, debido a su efectividad de predicción; como es el caso de los primeros autores en donde solo con el uso de algoritmos de optimización combinatoria logro un retorno de 0.0056 y una volatilidad muy baja de 0.0341, en el que se alcanzó un óptimo frente a las investigaciones consideradas en su estado del arte. Así también, para el caso de los segundos autores, se usaron redes neuronales para el manejo de series de tiempo en el que obtuvo como resultados un alto rendimiento justificado en una raíz del error cuadrático medio [RMSE] de 0.171 y un error absoluto medio [MAE] de 0.0948 lo cual se concluye con un alto rendimiento de la red frente a otros algoritmos. Otras investigaciones como la realizada por Bo (2023) en donde se usó algoritmos genéticos, logró mejorar el retorno de su portafolio a un valor de 0.2667 y una volatilidad de 0.0685, concluyendo el autor que el modelo propuesto supera al modelo de media-varianza de Markowitz y demuestra una estabilidad mayor al tener una volatilidad (riesgo) menor.

1.2. Formulación del problema

a. Problema General

¿Es posible optimizar un portafolio de inversión conformado por los activos que cotizan en la Bolsa de valores de Lima en el periodo 2024 usando algoritmos genéticos y una red neuronal artificial?

b. Problemas específicos

- ¿Cuál es la arquitectura de la red neuronal artificial para predecir la tendencia de los activos de capital que cotizan en la Bolsa de valores de Lima en el periodo 2024?
- ¿Cuál es el modelamiento de la función objetivo, individuos e hiperparámetros del algoritmo genético para la optimización del portafolio de inversión conformado por los activos que cotizan en la Bolsa de valores de Lima en el periodo 2024?

- ¿Cuál configuración del algoritmo genético es más eficiente para optimizar el portafolio de inversión conformado por los activos que cotizan en la Bolsa de valores de Lima en el periodo 2024?

1.3. Justificación de la investigación

Al conocer la posibilidad de invertir en el mercado de capitales, se evidencia una problemática respecto a que solo existen métodos determinísticos, y estos no contemplan la exploración combinatoria en el espacio de búsqueda total, por lo que se hace de vital importancia desarrollar herramientas metaheurísticas, que ayuden a los proyectistas e inversionistas en su toma de decisiones, al optimizar un portafolio de inversión. Esto permitirá mejorar el rendimiento de las unidades productoras frente a los excesos de liquidez que puedan generarse durante la etapa de pre-inversión o implementación de un proyecto, puesto que dentro de la problemática que afrontan tienen que evaluar de forma eficiente los medios por los cuales puedan obtener flujos positivos en los estados de resultados cuando se evalúe un proyecto o expansión del mismo, siendo estos excedentes de liquidez una fuente adicional de ingresos que podría mejorar las inversiones de la unidad productora.

En cuanto al aporte académico, con este proyecto se genera una base metodológica para el desarrollo de herramientas que ayuden a las decisiones económicas y financieras de personas naturales o jurídicas, entidades públicas; o cualquier interesado que quiera mejorar su condición económica mediante el uso de métodos algorítmicos de inteligencia artificial.

1.4. Objetivos de la investigación

a. Objetivo General

Diseñar un algoritmo genético y diseñar una red neuronal artificial para optimizar un portafolio de inversión conformado por los activos que cotizan en la Bolsa de valores de Lima en el periodo 2024

b. Objetivos Específicos

- Diseñar una red neuronal artificial para predecir la tendencia de los activos de capital que cotizan en la Bolsa de valores de Lima en el periodo 2024
- Modelar la función objetivo, representación del cromosoma de los individuos, y determinar los hiperparámetros de mejor respuesta al evaluar en el algoritmo para la

optimización del portafolio de inversión conformado por los activos que cotizan en la Bolsa de valores de Lima en el periodo 2024

- Evaluar las diversas configuraciones del algoritmo genético para la optimización del portafolio de inversión conformado por los activos que cotizan en la Bolsa de valores de Lima en el periodo 2024

1.5. Delimitación de la investigación

a. Delimitación temporal: Esta investigación tuvo como marco cronológico de desarrollo el periodo entre 2023 a 2024, por otra parte los datos obtenidos para el procesamiento de la red neuronal artificial tienen una temporalidad en un rango desde el 2 de enero del 2012 al 2 de enero del 2023, es decir el valor de las acciones de la bolsa de valores de Lima estuvo en este rango para el análisis exploratorio.

b. Delimitación espacial: La investigación fue desarrollada con un alcance de datos a nivel del geográfico de todo el Perú

c. Delimitación económica: Esta investigación se desarrolló con los recursos propios del investigador.

CAPITULO II: MARCO TEÓRICO CONCEPTUAL

2.1. Bases teóricas

2.1.1. Teoría de carteras de inversión

Según Álvarez García et al. (2004) frente a un caso donde existe acumulación de excedentes o de riqueza, se deben considerar 3 alternativas. La primera alternativa es en donde el agente racional o inversor decide quedarse bajo su tutela los excedentes, bajo la premisa que tiene total poder sobre estos, la segunda es llevar los recursos excedentes a un depósito financiero, en el cual lo que recibirá es un excedente o rendimiento sobre su inversión; y finalmente está la opción de invertir dichos recursos en activos que se encuentran en el mercado de valores.

La persona o ente inversor que ha decidido distribuir sus recursos tanto en el mercado de activos de valores, así como en el financiero, considera que esta decisión está vinculada con una medida de riesgo, debido a que los dos mercados en donde pondrá sus recursos trabajan bajo incertidumbre la cual no permite determinar con seguridad cuáles serán los resultados de la inversión. Respecto a esto se debe conocer que el agente que ha tomado la decisión de invertir se enfrenta a los siguientes tipos de riesgo:

- Riesgo de pérdida: Se da cuando el retorno de los activos es negativo
- Riesgo de oportunidad: Ocurre frente a la probabilidad de elegir activos que sean menos rentables que otros
- Riesgo de liquidez: Se da cuando los activos son devueltos a mercado y no se encuentra de forma oportuna compradores a cortos plazo
- Riesgo de inflación: Ocurre frente a cambios inesperados de los precios, afectando al rendimiento del activo comprado
- Riesgo de tasa de cambio: Ocurre cuando hay cambios en la valoración de activos internacionales

Frente a la presencia de riesgos, se ha formulado diversas interrogantes tales como: ¿Cómo se puede mitigar el riesgo?, ¿Cómo minimizar la probabilidad de pérdida en la cartera de inversiones formuladas por el inversor?

Dadas estas interrogantes, se han formulado diversas metodologías para darles respuesta, entre las cuales destacan:

Teoría de cartera de Markowitz: Según Markowitz (1952), el proceso de selección de una cartera de inversión se puede dividir en dos etapas. La primera etapa está basada en observación y experiencia, seguida del análisis de los rendimientos futuros de los activos disponibles. La segunda etapa involucra la evaluación de los factores relevantes que afectan los valores futuros y luego la selección de la cartera de inversión. Esta teoría se fundamenta en dos aspectos clave. El primero es maximizar los rendimientos descontados, también conocidos como rendimientos esperados o deseados, que son los comportamientos deseados por los inversores. El segundo aspecto considera que las variaciones en los rendimientos son indeseables. En este contexto, se menciona que los inversores tienden a preferir carteras diversificadas en lugar de no diversificadas, ya que la diversidad de activos existentes absorbe el riesgo que conlleva una cartera no diversificada al ofrecer correlaciones contrapuestas. La teoría de Markowitz se basa en las siguientes premisas:

1. Un inversionista puede determinar la rentabilidad y la distribución de probabilidad para cada acción o cartera de inversión.
2. La media de la distribución de rendimientos esperados expresa la rentabilidad esperada del activo o cartera.
3. La varianza o desviación estándar expresa el riesgo del activo o cartera.
4. El inversionista prefiere un título o cartera con mayor rentabilidad y menor riesgo.
5. La cartera eficiente para el inversionista se encuentra en el punto de tangencia entre las carteras eficientes y alguna de las curvas de indiferencia del inversionista.

Para contextualizar esto se puede definir matemáticamente los conceptos antes mencionados de la siguiente manera:

$$\begin{aligned}
 R &= \sum_{t=1}^{\infty} d_{it} r_{it} x \\
 &= \sum_{i=1}^N x \left(\sum_{i=1}^{\infty} d_{it} r_{it} \right) \\
 R &= \sum_{t=1}^{\infty} d_{it} r_{it}
 \end{aligned}$$

Donde R es el rendimiento descontado del valor i que representa un activo, entonces

t: Representa un periodo de tiempo

d_{it} : Tasa de descuento para el activo i en el periodo j

r_{it} : Rendimiento del activo i en el periodo j

N: Numero de activos totales

X: Cantidad de dinero invertid en el activo

Mientras que para la varianza que representa el riesgo, se puede determinar mediante lo siguientes elementos:

- σ_{ij} está relacionado con el coeficiente de correlación estadístico ρ_{ij}
- La covarianza entre R_i y R_j es igual al factor de correlación como factor multiplicador de las veces la desviación estándar de R_i y veces la desviación estándar de R_j , por lo tanto, definiéndose:

$$\sigma_{ij} = \rho_{ij}\sigma_i\sigma_j$$

Usando ese factor, la varianza de R_i es σ_{ii} entonces

$$v(R_i) = \sum_{i=1}^N \sum_{j=1}^N a_i a_j \sigma_{ij}$$

Sea R_i el rendimiento del valor i de algún activo. Sea μ_i el valor esperado de R_i ; σ_{ij} , sea la covarianza entre R_i y R_j y X_i el porcentaje de los activos del inversor que se asignan. El rendimiento (R) de la cartera en su conjunto es

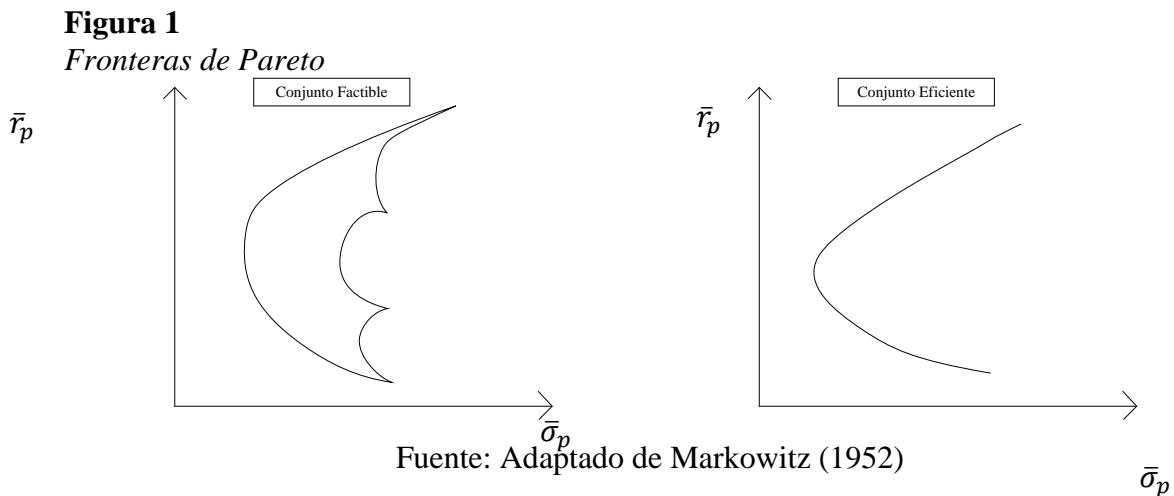
$$R = \sum R_i X_i$$

Los R_i (y en consecuencia R) son variables asociadas a la aleatoriedad. Las X_i no son variables aleatorias, sino que las fija el inversor. Así mismos asociados a estos tenemos dos restricciones

$$\sum X_i = 1$$

$$X_i \geq 1$$

Gráficamente se puede expresar la relación entre los rendimientos esperados (eje Y) y riesgos asociados (eje X) mediante la siguiente figura 1:



Capital Asset Pricing Model (CAPM): El modelo CAPM nace a partir de la motivación de encontrar una solución que minimice la cantidad de cálculos realizados en referencia a la teoría de Harry Markowitz, debido a que, si bien es sencilla de trabajar con 2 activos, se hace relativamente compleja cuando la cantidad de activos aumenta.(Pilbeam, 2005)

Pilbeam (2005) indica que la sugerencia de Sharpe en 1963 enfatiza en que las acciones tienden a moverse en diferentes grados en relación a los movimientos del mercado. Es decir, existe una correlación directa y lineal entre las acciones, y el mercado en general, postulando esta afirmación mediante la siguiente ecuación:

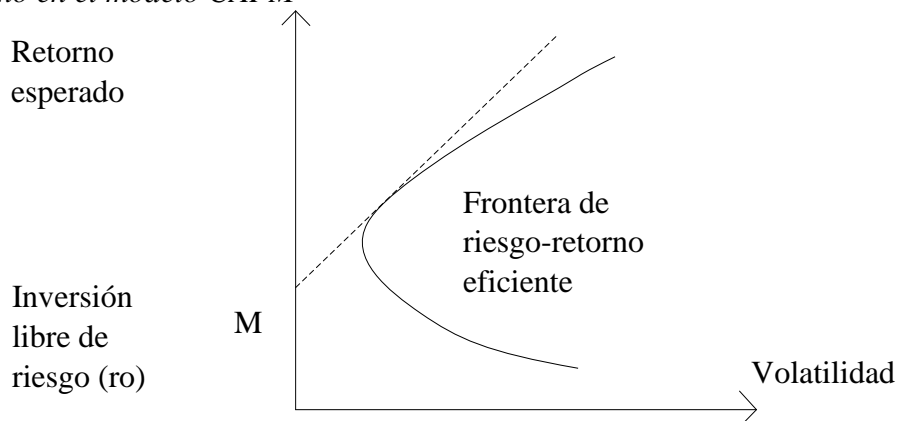
$$E(R_i) = \alpha_i + \beta_i E(R_m)$$

Donde R_i es una tasa de rendimiento esperado del activo i , $E(R_m)$ es la tasa esperada del mercado, mientras que α es una constante que tiene un rango entre los valores mencionados denominada tasa libre de riesgo y que usualmente se usan los bonos soberanos o títulos federales, ya que el riesgo a la insolvencia se elimina considerando que el gobierno determina impuestos para aumentar sus ingresos y pueda cumplir sus compromisos, y finalmente β es la sensibilidad que tiene el rendimiento del activo i en relación con el rendimiento del mercado. (Pilbeam, 2005), además los valores de beta pueden tener 3 interpretaciones, si es mayor a 1 el riesgo que no se puede diversificar de la inversión es mayor al del mercado, si es menor a 1 el riesgo no diversificable es inferior al riesgo del mercado y si es igual a 1 tiene una relación lineal con el riesgo del mercado (De Sousa, 2013)

Entre la contextualización del modelo CAPM, (De Sousa Santana, 2013) resalta que se debe considerar la opción teórica de invertir con una tasa libre de riesgo. Por lo que se hace necesario comprobar las opciones que están disponible en la frontera de riesgo-retorno y la inversión sujeta a un riesgo 0. Esto se muestra en la figura 2

Figura 2

Optimo en el modelo CAPM



Fuente: Adaptado de De Sousa (2013)

Observando la figura se puede identificar que la toma de decisión de invertir en un activo debe estar equilibrada en un portafolio de riesgo retorno eficiente.

Arbitraje Pricing Theory (APT): Esta teoría formulada por Stephen A. Ross en 1976 determina un retorno a partir de una inversión que presenta un riesgo definido, esto en base a una relación lineal de k elementos o factores, que a diferencia del CAPM permite incluir más de un factor de riesgo sistemático, por lo que el anterior modelo mencionado sería una representación específica de este modelo (De Sousa , 2013).

El riesgo sistemático generado en este modelo está representado por betas, además otro fundamento de este modelo es que el principio es la ausencia de arbitraje por lo que se permite generar un modelo multifactorial (Velasco & Mejía S, 2019). La representación matemática se expresa:

$$E(r_i) = r_f + \beta_{i1} * F_1 + \beta_{i2} * F_2 + \dots + \beta_{ik} * F_k$$

Donde los factores se definen:

$E(r_i)$: rentabilidad esperada del activo i

r_f = Rentabilidad del activo libre de riesgo

β_i = Coeficiente del activo i relacionado al factor 1

F = Es el elemento que representa la información macroeconómica

Cada beta antes mencionada o también definida como premio al riesgo, representa la rentabilidad obtenida de forma adicional por cada unidad de riesgo que el activo representa. (Velasco & Mejía, 2018)

Modelo de Fama y French: En la década de 1990 los autores Fama y French muestran un modelo en donde el rendimiento de los activos depende de tres factores de riesgo (Carmona & Vera, 2017). Estos elementos son:

- Riesgo de mercado: Según fama y French los elementos a invertir que tengan mayor exposición al riesgo de mercado tienen un comportamiento a mostrar mayor rendimiento.
- Tamaño de la empresa: Según los autores de esta teoría las acciones de empresas de menor tamaño tienden a mostrar mayores rendimientos que las empresas de mayor tamaño, esto debido a que las empresas pequeñas usualmente realizan sus actividades con mayor riesgo

- Ratio book to market: Es la relación que existe entre el valor contable de la empresa y su valuación en el mercado, siendo el indicador que, si esta ratio es más alta, los rendimientos esperados también serán más altos, siendo la premisa que el mercado subvalora las acciones de estas unidades.

Siguiendo la nomenclatura base de las fórmulas de portafolio se expone la ecuación para este modelo:

$$RA = rf + \beta(r_m - r_f) + SA * SMB + hA * HML + \alpha$$

Donde:

RA = es el rendimiento del activo A

rf = es la tasa libre de riesgo

$\beta(r_m - r_f)$ = Es el riesgo sistemático del mercado

SA = es el grado de exposición relacionado al tamaño de las empresas

hA = es el grado de cómo se exponen al riesgo relacionado a la liquidez

Teoría del portafolio de Black-Litterman: López Rojas et al. (2015) indican que este modelo representa una mejora al modelo de Markowitz, en lo que se resalta la presencia de n activos con un valor de capitalización M_i para cada uno, así mismo se menciona las ponderaciones de mercado de esos activos, y están representados por el vector $W = (w_1, w_2, w_3, \dots, w_n)$, considerando que la ponderación del activo i es:

$$w_i = \frac{M_i}{\sum_{i=1}^n M_i}$$

El coeficiente de aversión al riesgo (λ) es una constante la cual se define como:

$$\lambda = \frac{R_M - R_f}{\sigma_M^2}$$

Donde R_M es el retorno que da el mercado, R_f es una tasa libre de riesgo y el denominador σ_M^2 es la varianza del rendimiento del mercado. Así mismo se expresa que el exceso de retornos que están de forma implícita de equilibrio (Π) se expresa como:

$$\Pi = \lambda \Sigma W$$

Se debe considerar que este modelo es ajustado respecto a las creencias del inversor y la data histórica de rendimientos que da el mercado, por lo que se hace mención de un vector de excesos de retornos a partir de la incertidumbre generado por estos conceptos. Este vector $R = (R_1, R_2, R_3, \dots, R_n)$, en cada uno de sus elementos representa al retorno del activo

menos la tasa libre de riesgo. Por lo que el comportamiento de estos valores tiende a una distribución normal con los siguientes parámetros:

$$R \sim N(\mu, \Sigma)$$

Además μ es la representación que se aproxima a una distribución proporcional al producto de dos distribuciones normales. Siendo la representación de equilibrio:

$$\mu = N(\Pi, \tau\Sigma)$$

Donde τ es una constante que representa el nivel o grado de incertidumbre en relación con la precisión del cálculo de Π , por lo que, si el grado en que se da esta incertidumbre es alto, el valor τ se aproximará a 0 y en caso contrario será próximo a 1, además sus valores más usuales son 0,01 y 0,05; considerando estadísticamente que la incertidumbre de la media debe ser menor a la incertidumbre que representa la variable.

En relación con la segunda distribución está expone las expectativas que tiene el inversor en cuanto a los retornos del mercado, indicando que existe un conjunto de K expectativas que tienen relaciones lineales. Esta expectativa mencionada se determina como el rendimiento que se espera de un portafolio P_k , que tiene una distribución normal con un promedio q_k , y una desviación estándar representada por w_k . Las expectativas K con sus respectivos rendimientos esperados se expresan:

$$P^T = [p_1, p_2, \dots, p_k]$$

$$Q^T = [q_1, q_2, \dots, q_k]$$

Se tiene en cuenta que el vector p es la matriz que selecciona los activos de mercado que forman parte de las expectativas, y q es el vector que contiene estas expectativas que se representan de la siguiente manera bajo la formulación descrita:

$$P * \mu = Q + \varepsilon$$

Donde P es la matriz conocida $K \times n$; Q es el vector de expectativas que ya se han identificado y tienen un orden $K \times 1$, ε es el vector de condiciones aleatorias $K \times 1$ que tiene una media de 0 y una matriz diagonal que considera covarianzas representada por Ω , con distribución normal por lo que se deduce:

$$P * \mu \sim N(Q, \Omega)$$

En consideración Bernal (2013) menciona que los parámetros Ω es una matriz diagonal $K \times K$ con elementos ω_{ii} en la diagonal y cero en las demás posiciones. Mientras su valor de ω_{ii} sea mayor, se define un grado de confianza menor en los retornos Q .

Habiéndose encontrado estos elementos se puede definir el retorno que es la ponderación tanto de las expectativas del inversor como de los views, por lo que el primer factor del rendimiento es:

$$(\tau_{\Sigma})^{-1}\Pi + P^T\Omega^{-1}Q$$

Donde:

$(\tau_{\Sigma})^{-1}\Pi$ = es la ponderación de Π

$P^T\Omega^{-1}Q$ = es la ponderación sobre el valor de Q

Para complementar esta formulación se debe insertar las proporciones invertidas debido a que la suma ponderada debe ser igual a 1, para tal caso la ecuación anterior se debe multiplicar por:

$$((\tau_{\Sigma})^{-1} + P^T\Omega - 1P)^{-1}$$

Por lo anteriormente formulado se puede expresar finalmente el rendimiento de la siguiente manera:

$$E(R)_{B-L} = ((\tau_{\Sigma})^{-1} + P^T\Omega - 1P)^{-1} * (\tau_{\Sigma})^{-1}\Pi + P^T\Omega^{-1}Q$$

2.1.2. Complejidad computacional

El tema de la complejidad computacional se aborda desde los años 70, introduciendo un nuevo campo de investigación, la complejidad computacional en la que la primera interrogante es ¿qué tipo de problema se está abordando?, si este es fácil o no en consideración de los algoritmos con los que se pueda resolver. Si el problema es fácil existirá un algoritmo que lo resuelva en un tiempo polinomial, lo cual indica que la cantidad de operaciones aritméticas que se usan para su resolución es una función polinomial del tamaño del problema abordado. De manera contraria si la función no es polinomial, el algoritmo se considera no polinomial y su categorización es definida como problema difícil. La determinación del tipo de algoritmo por su complejidad, es el principal fundamento para conocer si este se puede resolver en un tiempo razonable o si su resolución requerirá años, o inclusive siglos (Vélez & Montoya, 2007).

Mientras que la complejidad algorítmica se refiere al tiempo en que demora un algoritmo para resolver un problema, por otra parte, la complejidad computacional es la que

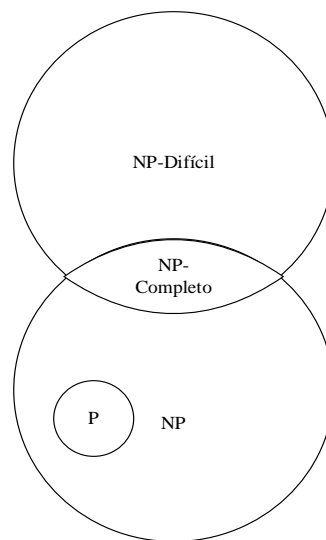
hace referencia a la cantidad de recursos computacionales que se necesitan para encontrar una solución a un problema. Fundamentalmente se refiere a elementos como el trabajo, la robustez, ductilidad de modelos matemáticos necesarios (Maldonado, 2013).

Categorización de problemas no polinomiales [NP]: Dentro de los problemas no polinomiales, el subconjunto de problemas que pueden ser resueltos en tiempo polinomial.

Los problemas NP-difíciles o duros, son considerados problemas que se pueden resolver en un tiempo polinomial no determinista, mientras los problemas NP completos que tienen naturaleza NP y NP-Duro al mismo tiempo. Por lo que una búsqueda de solución es factible a este tipo de problemas, no se considera que este se pueda encontrar en un tiempo razonable, ya que el proceso de crecimiento de este problema aumenta en forma proporcional al tiempo requerido para encontrar una solución factible.(Maldonado, 2013)

Los problemas NP Duros tiene sus mayores ejemplos en problemas de decisión, problemas de búsqueda o problemas de optimización. La conceptualización gráfica de los conjuntos de complejidad computacional se muestra en la figura 3

Figura 3
Conjuntos de complejidad computacional



Fuente: Adaptado de Granada & Santa (2016)

2.1.3. Metaheurística y la optimización combinatoria

Considerando que gran parte de problemas combinatorios son de naturaleza NP-completos, las metaheurísticas han demostrado su relevancia en este campo.

Las metaheurísticas son métodos que se fundamentan en conceptos algorítmicos cuyo objetivo es guiar los procesos heurísticos a evitar convergencia temprana, al explorar zonas o regiones del espacio de búsqueda que contienen soluciones de buena calidad, siendo muchas veces óptimas. Las metaheurísticas tienen la característica de poder encontrar soluciones generalizadas a un amplio abanico de problemas, por lo cual también tiene la denominación de “método heurístico de propósito general”.(Granada & Santa, 2016)

El procedimiento general de las metaheurísticas para resolver problemas de tipo combinatoria contempla 4 aspectos fundamentales:

1. Definición del espacio de búsqueda: Definido en relación al tamaño del problema y que tan complejo computacionalmente es este.
2. Codificación o representación del problema: En donde se contextualiza una arquitectura de alternativa de solución, sumado al conjunto de restricciones y una función objetivo a optimizar.
3. Población inicial: Por la naturaleza de las metaheurísticas para buscar una solución óptima en relación con soluciones aproximadas a esta en el espacio de búsqueda, se debe definir este conjunto de alternativas de inicialización, al cual se le denomina población inicial, el cual se genera a partir de una heurística o de forma aleatoria.
4. Definición de la estructura de vecindad: Este concepto hace referencia a los métodos o procedimientos a seguir para que las alternativas de cada interacción mejoren y se acerquen a un mejor valor de la función objetivo. De forma adicional la estructura que se escoja determina el tamaño del vecindario, es decir, define la cantidad de vecinos (soluciones alternativas) que se va a evaluar.

2.1.4. Algoritmos Genéticos

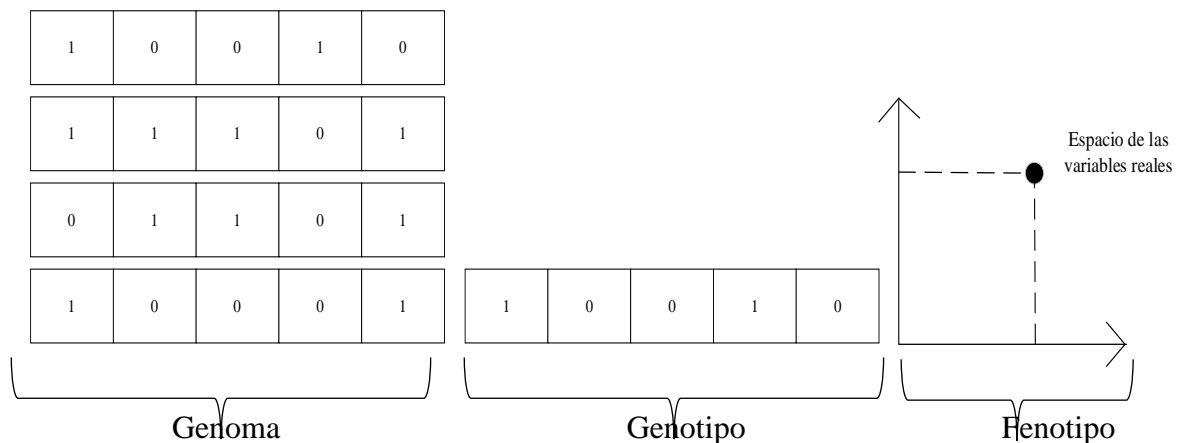
Los algoritmos genéticos son algoritmos de búsqueda basados en mecanismos de la selección natural y genética natural (Goldberg, 1989). Específicamente su fundamento se centra en la teoría evolucionista de Charles Darwin en donde indica que los individuos que mejor se adaptan al medio son aquellos cuya probabilidad de dejar descendencia es más alta, y cuyos genes pasarán a las generaciones próximas (Darwin, 1859). Con estas premisas, los algoritmos genéticos tienen como proceso principal la búsqueda estocástica guiada por la teoría de la evolución de los seres vivos para encontrar soluciones óptimas siendo eficientes y eficaces en su proceso.

Para los procesos en donde se debe encontrar una representación óptima numérica, los individuos son soluciones tentativas o potenciales al problema propuesto, y la forma en cómo se adaptan al medio se puede determinar por medio de la función a optimizar, llamada también fitness. Para un problema de maximización, el individuo demostrará su buen desempeño al encontrar un fitness alto, mientras que para un problema de minimización el desempeño se medirá óptimo cuando el valor del fitness sea bajo. (Gutiérrez et al., 2020)

Los arreglos que representan a los individuos que conforman una solución, son vectores que se denominan pseudo-cromosoma, y cada uno de los elementos que lo conforman se llaman alelos, estos definen las características de los individuos de acuerdo a los valores que tomen, y el cual se denomina genotipo, así mismo surge el concepto del fenotipo que es la representación real de la cadena cromosómica tal como se muestra en la figura 4.(Granada & Santa, 2016)

Figura 4

Conceptualización de población e individuos en algoritmos genéticos



Fuente: Granada & Santa (2016)

El genoma también definido como el grupo de alternativas que forman parte del espacio de búsqueda para el proceso de optimización. Esta población entra a ser procesada en el algoritmo genético durante una cantidad de iteraciones definidas por el diseñador del algoritmo, llamadas generaciones, las cuales tendrán como restricción un número definido o un criterio específico de parada. Para cada generación los individuos se someterán a un conjunto de operaciones en donde se reproducen, sobreviven y desaparecen lo que garantiza que en el proceso el tamaño de la población sea constante. Además, que para el proceso de selección es necesario verificar una métrica de calidad del individuo, que está relacionado

con un valor que se obtiene de una función objetivo y a que se cumplan una serie de restricciones. El procedimiento para calcular esta métrica de calidad recibe el nombre de función de adaptación.(Granada & Santa, 2016)

Para que el algoritmo encuentre soluciones que sean mejores a las anteriores al evaluarse mediante la condición de adaptación, es imprescindible modificar a los individuos mediante el uso de operadores de búsqueda local, en los que para un algoritmo genético se clasifican en:

- Operadores de selección: cuya finalidad es duplicar las soluciones con buen desempeño o calidad de individuo y aplicar pena de muerte a las soluciones con una calidad deficiente, con la restricción que la población se mantiene constante. Entre los métodos más usados está la selección por torneo, selección proporcional y selección por rango.
- Operadores de variación: a los que se asocia la creación de nuevas soluciones, lo que implique aumentar la diversidad de la población durante el proceso de optimización, que como consecuencia tiene generar descendientes únicos, bajo la teoría Darwinista de herencia de las mejores características de los padres.

Estos operadores tienen 2 subcategorías:

- El operador de cruzamiento o recombinación el cual mediante la combinación genética de 2 individuos de la población inicial crea descendimientos
- El operador de mutación que cambia al individuo por otro

Estos operadores buscan encontrar de forma intensiva el equilibrio de dos funciones en su búsqueda de individuos óptimos:

- La exploración del espacio de búsqueda en la cual el objetivo es encontrar buenas áreas y que tengan una calidad aceptable orientándose a obtener un óptimo global
- La explotación de las áreas de buena calidad encontradas, ya que al concentrar la búsqueda en esta región se podría obtener un óptimo

Aquí se hace énfasis en los procesos de búsqueda aleatoria para explorar zonas en el espacio de búsqueda que contengan individuos de calidad, sin embargo, el tiempo para encontrarlo no es razonable en muchas ocasiones. En contraste un operador de búsqueda local determinístico (Goloso) posee la capacidad de hallar un óptimo en un espacio de búsqueda, pero con la problemática de encontrar en un óptimo local, por lo que se debe encontrar el equilibrio.

Mecanismo de selección: Jiménez (2016) menciona que en esta etapa se transmite y conserva las características de los individuos más representativos y que pasarán a lo largo de las generaciones, y el mecanismo para lograr esto es que los individuos mejor adaptados a la función de evaluación, posean más probabilidades de reproducirse, así también el autor menciona que es necesario agregar un elemento aleatorio que habilite a los individuos que no estén bien adaptados al entorno, reproducirse si es que contienen información valiosa, cumpliendo con el factor de diversidad de la población (p. 18).

Selección por ruleta: Es el método más usado, y que tiene por característica que la probabilidad del individuo de reproducirse es proporcional al valor resultante, al ser evaluado por la función de evaluación, es decir a su capacidad de adaptarse al entorno.

Selección por Ranking: Este método ordena la población generada, cuyo valor de su función de calidad sea mejor hasta la de menor valor representativo. El orden de este ranking define la probabilidad de ser seleccionado, por lo que los individuos en los primeros lugares tendrán mayor probabilidad.

Selección por torneo: En este mecanismo se selecciona de forma aleatoria un conjunto de n individuos, y se define un número aleatorio en el rango de 0 a 1. Si el valor del número aleatorio es menor al umbral determinado, por ejemplo 0.75 se escoge al individuo con mejor valor de función de adaptación, y si este número es mayor, se selecciona al individuo con la adaptación más baja.

Reproducción o Cruza: Este operador permite a los individuos que pasaron por el proceso de selección, reproducirse y obtener nueva descendencia. Además, se debe considerar que estos descendientes tengan mejores características que los padres debido a que el mecanismo de selección tiene como propósito escoger a los individuos que mejor se adapten al entorno (Jiménez, 2016, p. 21).

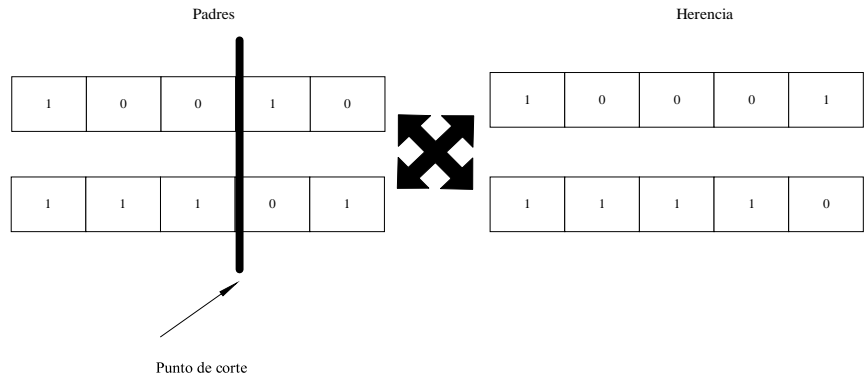
cruzamiento por recombinación de cadenas binarias: en su forma más práctica de aplicación es escoger de forma aleatoria dos cromosomas (individuos) los cuales previamente fueron escogidos por el operador de selección, y entre ellos cambiar porciones de genes (ver figura 5). (Granada & Santa, 2016)

Por lo general este operador tiene las siguientes propiedades:

- El cruzamiento de dos padres iguales, tiene como resultados un descendiente idéntico a los padres

- Dos individuos con características similares, en el espacio de búsqueda tendrán como descendiente un individuo similar a ellos

Figura 5
Proceso de cruzamiento en algoritmos genéticos



Fuente: Adaptado de Granada & Santa (2016)

Cruza de n puntos: Se escoge de forma aleatoria n posiciones en las cadenas de los padres y se realiza el cruzamiento intercambiando los genes en las posiciones de los progenitores tal como se muestra en la figura 6 (Jiménez, 2016, p. 22).

Figura 6
Representación de cruce de n puntos

Padre 1	0	1	1	0	1	0	1	0	1
Hijo 1	0	1	1	0	1	0	0	0	1

Padre 2	1	1	1	0	1	0	0	0	1
Hijo 2	1	1	1	0	1	0	1	0	1

Fuente: Adaptado de Jiménez (2016)

Cruza uniforme: Esta técnica tiene un enfoque diferente, ya que cada gen de la descendencia tiene la misma probabilidad de pertenecer a alguno de los padres. La forma más usual de implementar esta técnica es mediante el uso de una máscara conformada por valores binarios, en donde si el valor dentro de esta cadena es 1, se escoge el mismo valor del locus del primer padre y en caso sea 0, se escoge el valor del locus del segundo padre tal como se muestra en la figura 7 (Jiménez, 2016, p. 22).

Figura 7
Representación de cruce uniforme

Máscara	0	1	0	0	1	0	1	0	1
Padre 1	9	5	3	1	2	4	6	8	7

Padre 2	7	1	4	9	3	8	5	6	2
---------	---	---	---	---	---	---	---	---	---

Hijo 1	-	5	-	-	2	-	6	-	7
Hijo 2	7	-	4	9	-	8	-	6	2

Hijo 1	1	5	4	9	2	3	6	8	7
Hijo 2	7	5	4	9	3	8	1	6	2

Fuente: Adaptado de Jiménez (2016)

Cruza por emparejamiento parcial: Dados dos progenitores, lo que indica este operador es que copia una cadena de valores de uno de los padres y lo traslada a las mismas ubicaciones del segundo progenitor, quedando los valores no intercambiados sin alteración del segundo progenitor en el mismo orden en el que se encontraban, esto se muestra en la figura 8 (Jiménez, 2016, p. 24).

Figura 8

Representación de cruce por emparejamiento parcial

Padre 1	1	3	6	4	2	5	8	7	
Padre 2	2	7	8	3	1	5	4	6	
Hijo 1				3	1	5			
Hijo 2				4	2	5			
Hijo 1			6	3	1	5	8	7	
Hijo 2		7	8	4	2	5		6	
Hijo 1	2	4	6	3	1	5	8	7	
Hijo 2	1	7	8	4	2	5	3	6	

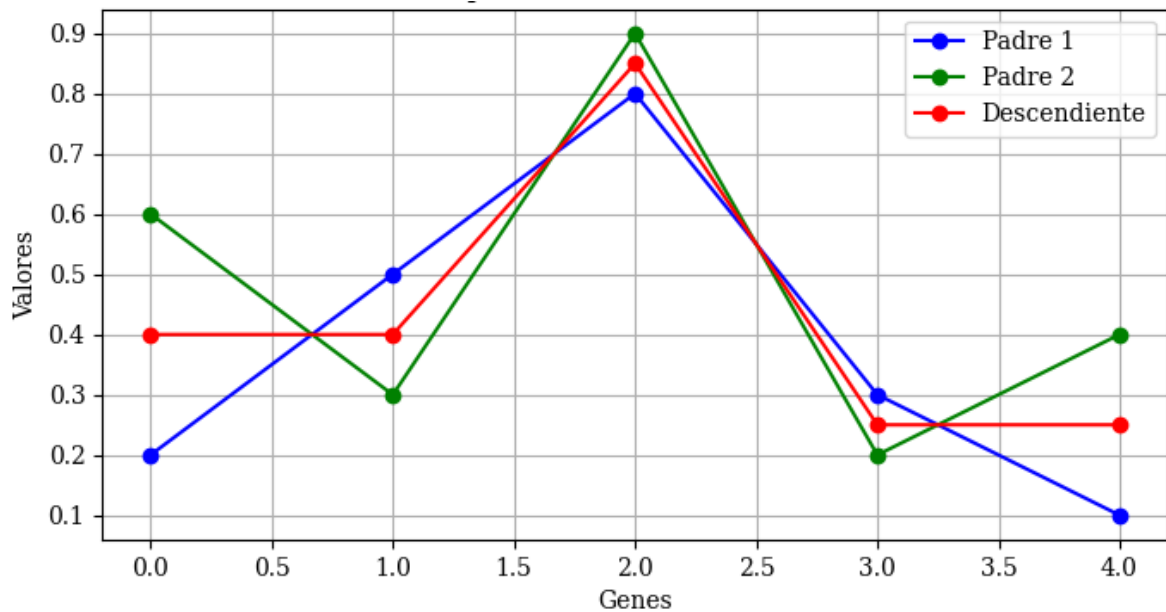
Fuente: Adaptado de Jiménez (2016)

Una forma análoga se este proceso de cruzamiento es el llamado Cxblend en donde cada individuo obtiene características de sus progenitores en una proporción lineal, esta proporción está dada por un factor alfa, asegurando que el individuo descendiente siempre este en su representación de vector entre los dos padres, y su representación algebraica es:

$$D_i = P_{1i} + \alpha(P_{2i} - P_{1i})$$

El comportamiento de este mecanismo se muestra en la siguiente figura 9:

Figura 9
Operación de cruzamiento CxBlend



Cruza en orden: En este mecanismo se toma en cuenta tomar un conjunto de valores del padre 1 y se preserva el orden relativo de los genes del resto, sin alterar la estructura cromosómica del padre 2, esto se muestra en la figura 10 (Jiménez, 2016, p. 25).

Figura 10
Representación de cruce en orden

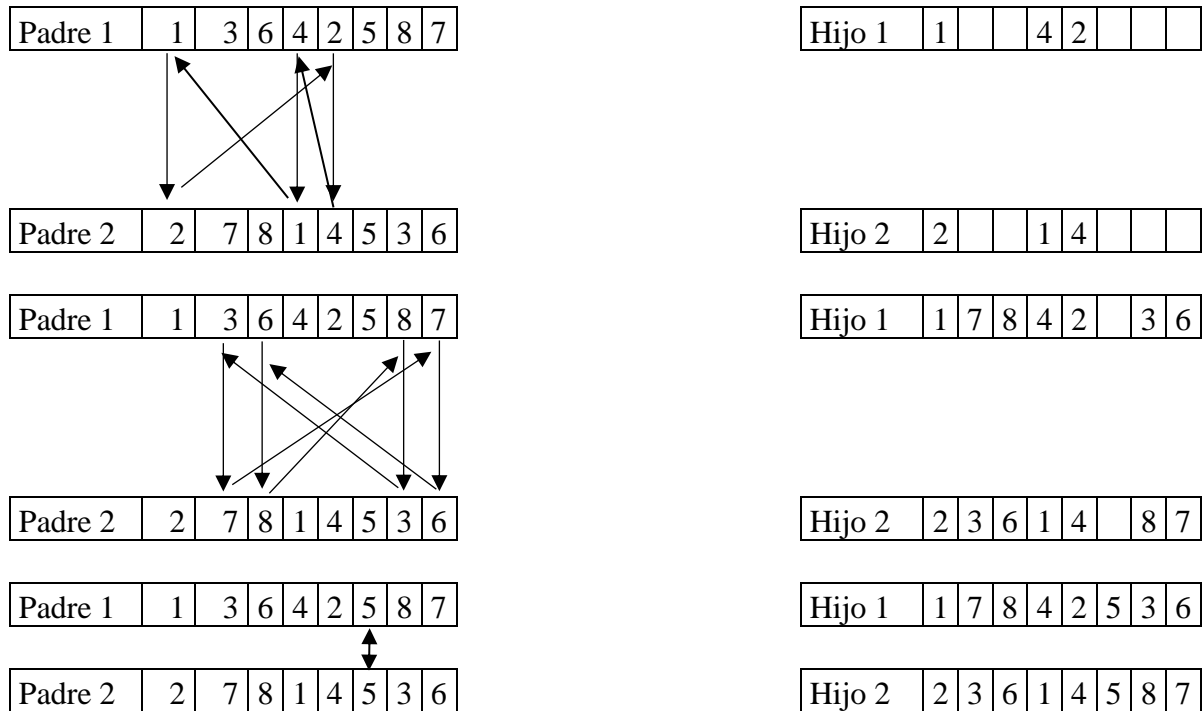
Padre 1	1	3	6	4	2	5	8	7
Padre 2	2	7	8	3	1	5	4	6
Hijo 1				4	2	5		
Hijo 2				3	1	5		
Hijo 1				4	2	5	7	8
Hijo 2				3	1	5	6	4
Hijo 1	3	1	6	4	2	5	7	8
Hijo 2	2	8	7	3	1	5	6	4

Fuente: Adaptado de Jiménez (2016)

Cruza por ciclos: La idea fundamental es conservar la posición en el cual los genes y alelos se presentan. Los descendientes son creados a partir de la selección de ciclos de los progenitores, tal como se muestra en la figura 11 (Jiménez, 2016, p. 26).

Figura 11

Representación de cruce por ciclos

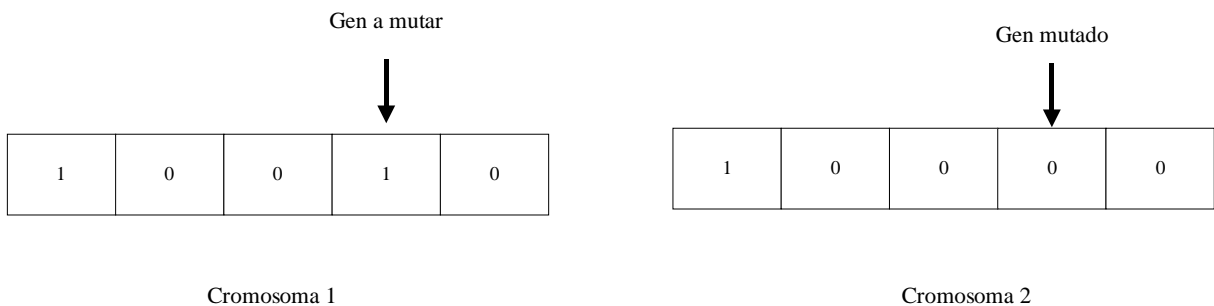


Fuente: Adaptado de Jiménez (2016)

Mutación: Este operador tiene por finalidad alterar a un individuo de forma local, en donde mediante un proceso estocástico modifica a un alelo de un pseudo-cromosoma en un gen aleatorio tal como se muestra en la figura 12. Para este operador se debe tener en cuenta que no siempre genera una solución mejor a la anterior, pero si la probabilidad de ocurrencia es baja, la exploración en el espacio de búsqueda se acota. Esta probabilidad mencionada oscila entre 0.01 y 0.1 para cada individuo y su denominación es tasa de mutación t_m (Granada & Santa, 2016).

Figura 12

Operación de mutación



Fuente: Adaptado de Granada & Santa (2016)

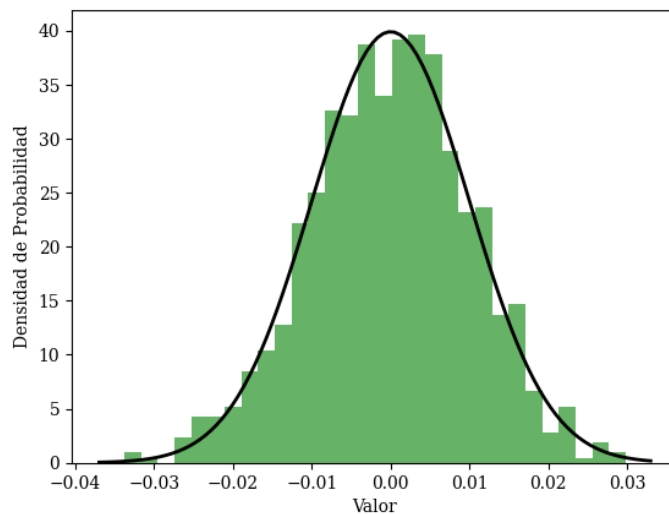
Mutación Gaussiana: consiste en que a partir de un cromosoma x en donde se selecciona un gen para la mutación i se le aplica la distribución normal N de media μ_i y desviación estándar s , para el cromosoma p j -ésimo del gen seleccionado se genera el cromosoma c representado por la siguiente ecuación:

$$x'_i = x_i + N(0, \sigma)$$

Su representación gráfica de muestra a continuación en la siguiente figura 13:

Figura 13

Mutación Gaussiana

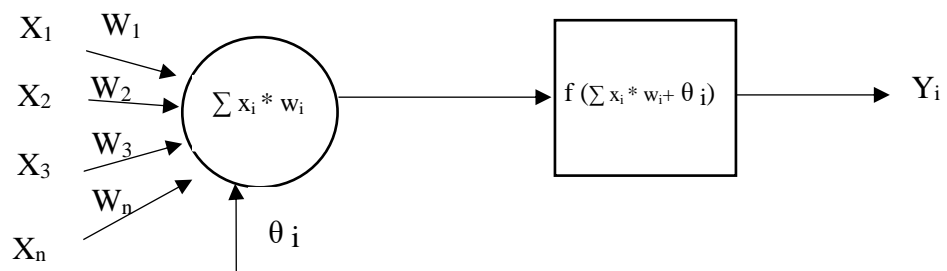


2.1.5. Redes neuronales artificiales

La neurona artificial: También conocida como célula o autómata, es el elemento que contiene un estado en su interior, el cual se denomina umbral, nivel o grado de activación, y que a partir de señales que la estimulan puede cambiar su estado (Isasi & Galván, 2004). A modo gráfico a continuación se muestra una neurona en la Figura 14.

Figura 14

Neurona artificial



Fuente: Adaptado de Vorobioff et al. (2022)

Donde:

X_i = vector que representa las entradas de la red

W_i = vector que representa los pesos de una determinada capa de la red

θ_i = Umbral de activación de la red

$f(\sum x_i * w_i)$ = Función de activación cuyo argumento el producto de los pesos por las entradas de una determinada capa de la red

Y_i = Salida de la red

La estructura mostrada, contiene un conjunto de conexiones, o pesos sinápticos que son los que se ajustan para definir el comportamiento de la neurona, y cuya naturaleza es de activación o inhibición; también se contempla una función de activación, un umbral que define el límite en cual la neurona se activa (Serrano et al., 2010). Por otra parte Vorobioff et al. (2022) detalla el flujo de los datos, en donde las conexiones con otras neuronas son definidas bajo el concepto biológico de sinapsis, mientras que la unidad de procesamiento de producto y suma de los pesos y las entradas, asume el nombre de cuerpo de la neurona, y el resultado de la operación anterior pasa por una función de activación, la cual es la encargada de generar fronteras de decisión no lineales dependiendo del tipo de función seleccionada; finalmente la salida de la función de activación pasa por el axón para ser entrada a otras neuronas artificiales o la salida final de la red según sea el caso o tipo de estructura.

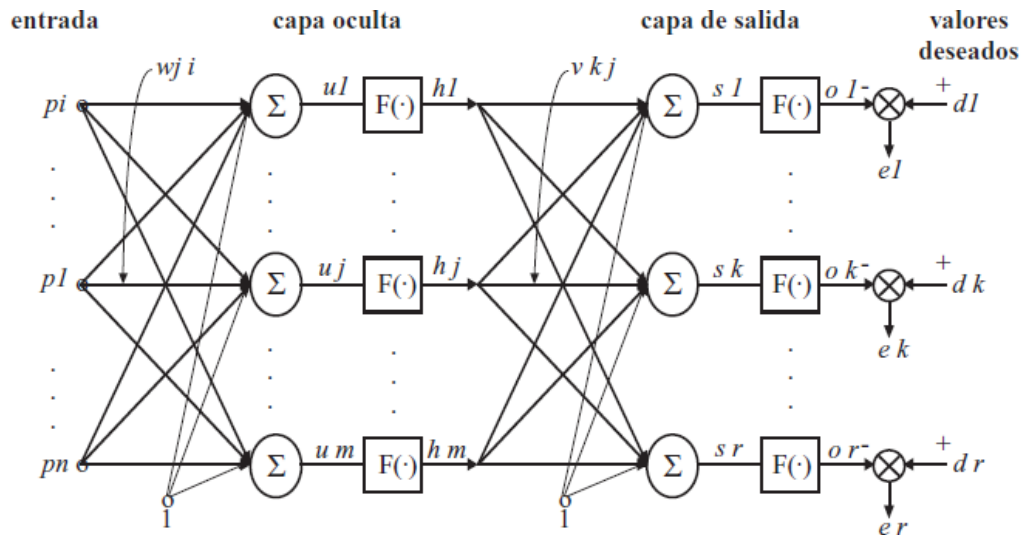
Características de las redes neuronales: Según Basogain (1998) estos modelos se inspiran en la arquitectura morfológica de las redes neuronales biológicas, en las que su funcionamiento también es análogo en los procesos más básicos. Así mismo sus características se pueden expresar de la siguiente manera:

- Aprenden a partir de la experiencia e interacción con su entorno, ajustando sus salidas para que tengan consistencia
- Generalizan en función a ciertos hiper parámetros configurados para su aprendizaje
- Abstraen, es decir pueden identificar o agrupar ciertos parámetros del entorno de su aprendizaje sin haber caracterizado previamente estas cualidades.

Estructura básica de una red neuronal: La estructura fundamental que modela la interconexión generada entre neuronas, se define como red multicapa (figura 15). En la estructura de entrada inicial, están las células receptoras, por las cuales ingresan vectores de datos que son los valores a los cuales se le aplicará un procesamiento en las capas

intermedias, las cuales se denominan capas ocultas para finalmente encontrarse con la capa de salida. (Isasi & Galván, 2004)

Figura 15
Estructura de red neuronal artificial



Fuente: Ponce Cruz (2010)

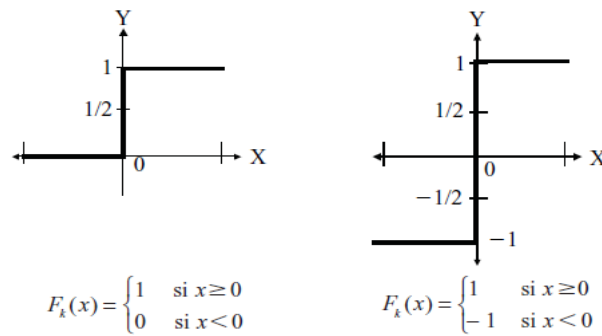
Funciones de activación: Se le denomina función de activación F_k a la regla que determina el efecto de la entrada ponderada total de la unidad neuronal k (Ponce Cruz, 2010). Su representación en forma de función se define:

$$y(t + 1) = F_k \left(\sum_j \omega_j(t) x_j(t) + \theta(t) \right)$$

Algunas de las funciones de activación más usadas son:

Función escalón: Se relaciona con las neuronas artificiales del tipo binarias, en el caso de que la suma de las entradas es mayor o igual al umbral, el resultado o salida de la función es 1, en caso contrario es 0 o -1 según este configurada la función (ver figura 16).

Figura 16
Función escalón



Fuente: Ponce Cruz (2010)

Función lineal o mixta: La función lineal, también llamada identidad se representa mediante la función $F_k(u) = u$. Para el caso de las neuronas que tienen esta función las reglas que siguen las salidas son:

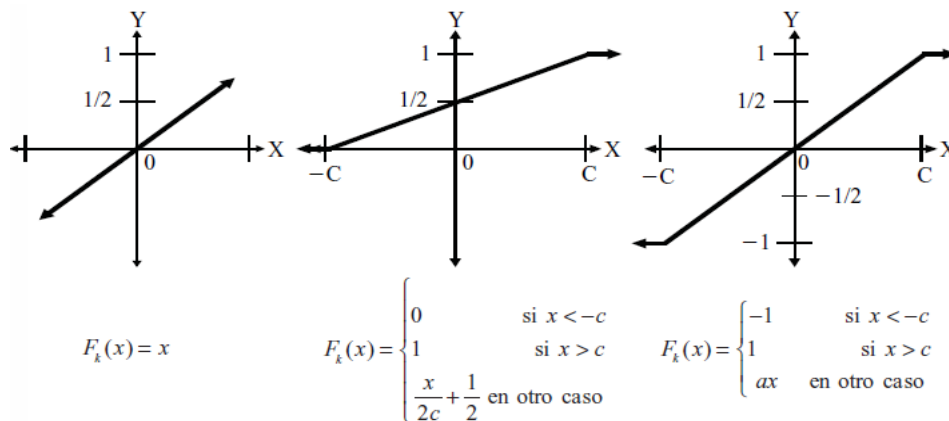
Si la suma de las señales \leq al límite inferior entonces la activación es 0 o -1

Si la suma de las señales \geq al límite superior entonces la activación es 1

Si la suma de las señales esta entre los limites entonces la activación es 1

La representación gráfica se puede observar en la figura N° 17:

Figura 17
Función lineal

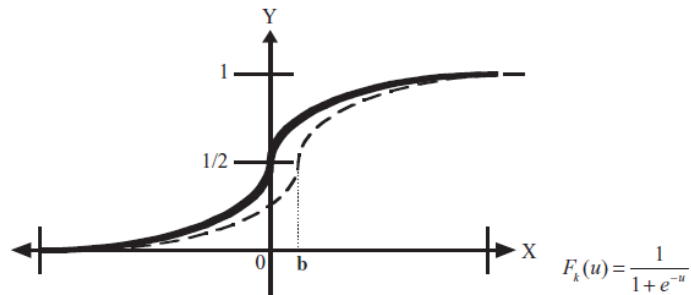


Fuente: Ponce Cruz (2010)

Función sigmoidal: En esta función hay dos asíntotas bien definidas, lo que hace que las salidas este comprendidas en la mayoría de casos en la zona alta o baja de la función. Esta función ha demostrado su utilidad al tener su derivada cercana a 0 para valores grandes sean positivos o negativos, siendo utilizada muchas veces en las reglas de aprendizaje (ver figura 18).

Figura 18

Función Sigmoide

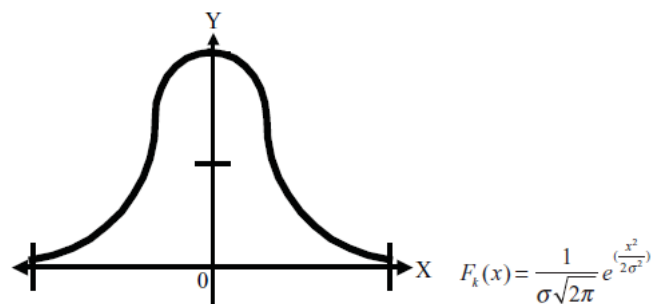


Fuente: Ponce Cruz (2010)

Función de Gauss: Han demostrado su utilidad para el reconocimiento de patrones en un solo nivel de neuronas, siendo más eficientes que el tipo de función sigmoide (ver figura 19).

Figura 19

Función de Gauss



Fuente: Ponce Cruz (2010)

Aprendizaje de las redes neuronales artificiales: El aprendizaje de una red neuronal es el aspecto más importante dentro de este tipo de algoritmos, y cumple un papel fundamental en el tipo de problema que se resolverá. El aprendizaje se basa en el tipo de ejemplos que disponga la red para su entrenamiento, bajo este concepto estos ejemplos deben cumplir dos requisitos fundamentales, que sea significativos, es decir que haya suficientes ejemplos para que la red aprenda a generalizar los resultados a cualquier tipo de dato; y la otra condición es que sea representativo, es decir que el set de datos este diversificado en una amplia variedad de ejemplo de muchas categorías.

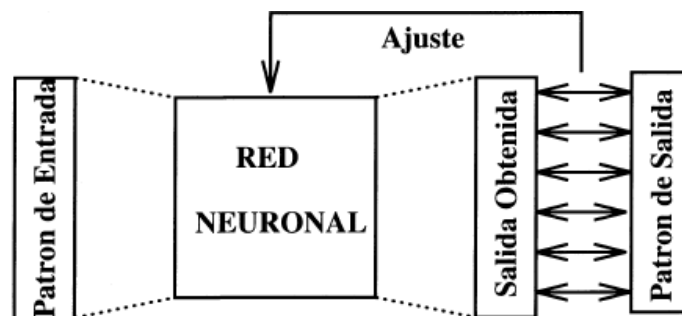
El aprendizaje de la red neuronal artificial estrictamente hablando en cuanto a su topología se refiere a la determinación precisa de los pesos sinápticos de todas sus conexiones con la finalidad de habilitar a la red a que resuelva un problema determinado. El proceso general consiste en introducir de forma gradual los datos del set de entrenamiento e ir modificando sus pesos teniendo en cuenta una regla o norma de aprendizaje, y considerando un criterio de convergencia.

Tipos de aprendizaje:

Aprendizaje supervisado: En este tipo de aprendizaje el set de datos tiene dos tipos de atributos, los propios datos e información relativa a la solución del problema, por lo que el procesamiento que se realiza en un primer momento de forma unidireccional verifica las salidas o respuestas de la red y las compara con la salida que debió tener la red y a partir de esto se modifican los pesos hasta minimizar el error de la red en la mayor medida posible. El esquema general de este tipo de aprendizaje se puede expresar según la figura 20.

Figura 20

Esquema de aprendizaje supervisado

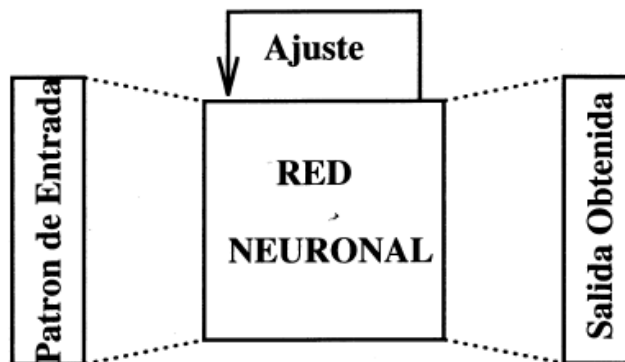


Fuente: Isasi & Galván (2004)

Aprendizaje no supervisado: En este tipo de aprendizaje solo se tienen los datos del set de entrenamiento, y no existe alguna regla referencial para el proceso de aprendizaje, por lo que la red ira aprendiendo a partir de los mismos datos internos, características, regularidades o redundancias para así modificar sus pesos, por esta razón este tipo de aprendizaje da lugar a los sistemas autoorganizados (Isasi & Galván, 2004). El esquema del aprendizaje no supervisado se puede observar en la figura 21.

Figura 21

Esquema de aprendizaje no supervisado



Fuente: Isasi & Galván (2004)

Para determinar si es que la red neuronal artificial da como respuesta salidas adecuadas, se divide el set de entrenamiento en dos subconjuntos, que tendrán por denominaciones, set de entrenamiento y de validación, el primero se utiliza para modificar y ajustar el valor de los pesos, mientras que el de validación se usara para ajustar el error de la red, de manera que para medir la precisión y eficiencia de la red se utilizaran datos que la red no conoce, por no haber sido utilizados en el proceso de aprendizaje.(Isasi & Galván, 2004)

De acuerdo a lo mencionado anteriormente el proceso de aprendizaje se resume de la siguiente manera:

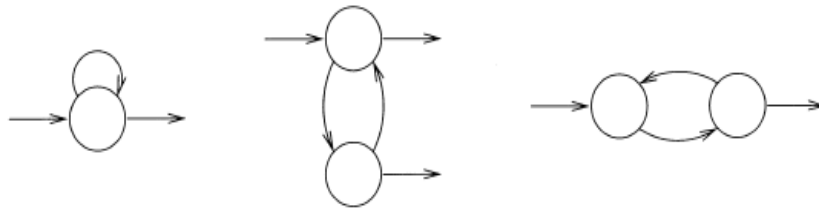
1. Asignar valores a los pesos de forma aleatoria
2. Procesar el aprendizaje de la red con los datos de entrenamiento modificando los pesos
3. Usar los datos de validación para evaluar el error de la red
4. Si el valor de error esta fuera de un rango determinado como umbral entonces se procederá a regresar al paso 2
5. Finalizar el proceso de aprendizaje y dar como resultado la topología de la red.

Redes neuronales recurrentes: Isasi & y Galván (2004) mencionan que las redes neuronales recurrentes son aquellas cuya característica principal es que entre las neuronas se crean bucles mediante las conexiones denominadas recurrentes, por lo que se podrá identificar conexiones de neuronas hacia ellas mismas, conexiones entre neuronas en una

misma capa, o conexiones de neuronas hacia la capa anterior tal como se muestra en la figura 22.

Figura 22

Ejemplo de conexiones recurrentes



Fuente: Isasi & Galván (2004)

Si se considera en el diseño conexiones recurrentes en donde se crean bucles, la activación de una neurona con este tipo de conexiones no dependerá solo de las activaciones de las neuronas de la capa anterior, sino también del estado de activación de cualquier conexión hacia la neurona de entrada, incluyendo su propia activación. Teniendo en cuenta lo anterior se debe considerar en este tipo de redes neuronales la variable tiempo o estado de neurona la cual se define de la siguiente manera:

$$a_i(t + 1) = f_i \left(\sum_j w_{ji} a_j(t) \right)$$

Donde j tiene variación en el grupo de todas las neuronas que se conectan a la neurona i.

Como se mencionó anteriormente, se debe considerar la variable tiempo en las activaciones de las neuronas recurrentes, esto hace que este tipo de red tenga un comportamiento dinámico o temporal. Teniendo en cuenta esto el aprendizaje de esta red y su forma de procesamiento se puede definir de las siguientes dos maneras:

Evolución de las activaciones de la red para alcanzar un estado estable: La finalidad de este procedimiento es evolucionar la red desde un estado inicial hasta un estado final en el que las activaciones de todas las neuronas pertenecientes de la red no se modifiquen, es decir, la salida representará el estado estable.

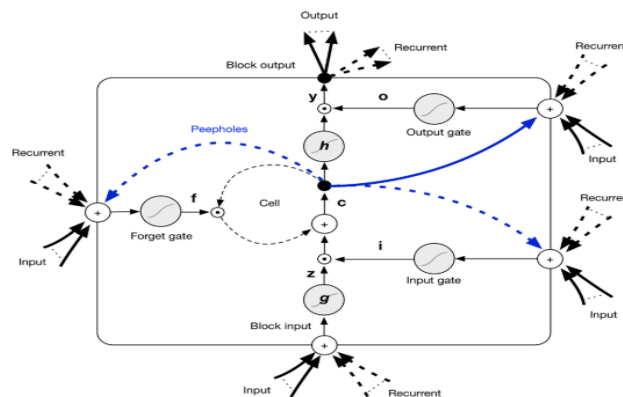
Evolución de las activaciones de la red en modo continuo: En este contexto cada momento en términos de tiempo se encuentra explícito en la salida de la red, el cual tiene una relación con la entrada en el momento inmediato anterior. De acuerdo con este tipo de conceptualización de la red el aprendizaje se puede generar a partir de dos formas distintas-

Primero el aprendizaje por épocas se considera un intervalo de tiempo, es en este intervalo en donde la red evoluciona y al alcanzar el punto final del intervalo, los pesos de la red se adaptan. Y la segunda forma es el aprendizaje en tiempo real o continuo donde la regla de aprendizaje por la cual se modifican los pesos se aplica en cada instante de tiempo.

Redes recurrentes Long Short Time Memory [LSTM]: Debido a las características de las redes neuronales recurrentes, que incluyen conexiones que retroalimentan la red, estas son ampliamente utilizadas en el procesamiento de series temporales o patrones dinámicos, donde la salida depende del estado previo de la red. Según Patterson y Gibson (2017), las redes LSTM son la variante más popular de las redes neuronales recurrentes, y fueron introducidas por Hochreiter y Schmidhuber en 1997. La característica crítica de las LSTM es su unidad de memoria y las compuertas asociadas (como la de olvido y la de entrada). Estas compuertas regulan el contenido de la memoria, lo que permite que la información se retenga a lo largo de múltiples pasos de tiempo. Si ambas compuertas están cerradas, el contenido de la memoria permanece inalterado de un paso de tiempo al siguiente. La estructura de compuertas facilita la retención de información a lo largo del tiempo, lo que a su vez permite que los gradientes se propaguen a través de varios pasos de tiempo. Esto soluciona el problema del desvanecimiento del gradiente que afecta a la mayoría de los modelos de redes neuronales recurrentes.

Unidades LSTM: Las unidades en las capas de las Redes Neuronales Recurrentes (RNN) son una adaptación de la clásica neurona artificial. Cada unidad de una Red Neuronal de Memoria a Corto y Largo Plazo (LSTM) tiene dos tipos de conexiones: las conexiones que provienen del paso de tiempo anterior (las salidas de esas unidades) y las conexiones que provienen de la capa anterior en la arquitectura de la red (ver figura 23).

Figura 23
Celda LSTM



Fuente: Patterson & Gibson (2017)

Donde los componentes en una unidad LSTM son:

- Tres compuertas:
 - Compuerta de entrada (compuerta de modulación de entrada)
 - Compuerta de olvido
 - Compuerta de salida
- Entrada de bloque
- Celda de memoria (el carrusel de error constante)
- Función de activación de salida
- Conexiones

Hay tres unidades de compuerta, que aprenden a proteger la unidad lineal de señales engañosas:

La compuerta de entrada protege la unidad de eventos de entrada irrelevantes.

La compuerta de olvido ayuda a la unidad a olvidar los contenidos de memoria anteriores.

La compuerta de salida expone (o no) los contenidos de la celda de memoria en la salida de la unidad LSTM.

La salida del bloque LSTM está conectada recurrentemente de vuelta a la entrada del bloque y a todas las compuertas del bloque LSTM. Las compuertas de entrada, olvido y salida en una unidad LSTM tienen funciones de activación sigmoideas para restringir sus valores entre 0 y 1. La función de activación de entrada y salida del bloque LSTM, generalmente, es una función tangente hiperbólica (tanh).

Utilizando la notación de Greff et al., la pasada hacia adelante (forward pass) de una capa LSTM se realizaría de la siguiente manera:

$$z^t = g(W_z x^t + R_z y^{t-1} + b_z) \dots \dots \dots \text{Bloque de entrada}$$

$$i^t = \sigma(W_i x^t + R_i y^{t-1} + P_i \odot C^{t-1} + b_i) \dots \dots \dots \text{Compuerta de entrada}$$

$$f^t = \sigma(W_f x^t + R_f y^{t-1} + P_f \odot C^{t-1} + b_f) \dots \dots \dots \text{Compuerta de olvido}$$

$$C^t = (i^t \odot z^t + f^t \odot C^{t-1}) \dots \dots \dots \text{Celda de estado}$$

$$o^t = \sigma(W_o x^t + R_o y^{t-1} + P_o \odot C^t + b_o) \dots \dots \dots \text{Compuerta de salida}$$

$$y^t = (o^t \odot z^t + h(C^t)) \dots \dots \dots \text{Bloque de salida}$$

Donde:

x^t : Vector de entrada en el momento t

W : Matrices de peso de entrada rectangular

R : Matrices de peso recurrente cuadradas

P : Vector de peso de compuertas

b : Vector de sesgos

Capas LSTM: Una capa básica de una red neuronal recurrente (RNN) acepta un vector de entrada x (que puede variar en longitud) y produce una salida y . La salida y está influenciada tanto por la entrada x como por el historial de todas las entradas anteriores. La influencia del historial de entradas se lleva a cabo a través de las conexiones recurrentes dentro de la capa. Además, la RNN tiene un estado interno que se actualiza cada vez que se ingresa un nuevo vector a la capa. Este estado consiste en un solo vector oculto, que captura información sobre el contexto de las entradas anteriores y permite que la red capture dependencias temporales en los datos de entrada.

Entrenamiento LSTM: Las redes LSTM utilizan aprendizaje supervisado para actualizar los pesos en la red. Durante el entrenamiento, se procesa un vector de entrada a la vez en una secuencia de vectores. Estos vectores contienen valores reales y se convierten en secuencias de activaciones de los nodos de entrada en la red. Cada unidad no de entrada calcula su activación actual en cualquier momento dado. Esta activación se determina mediante una función no lineal de la suma ponderada de las activaciones de todas las unidades desde las cuales recibe conexiones.

Para cada vector de entrada en la secuencia de entrada, el error se calcula como la suma de las desviaciones entre todas las señales objetivo y las activaciones correspondientes calculadas por la red. Este error se utiliza para ajustar los pesos de la red durante el proceso de entrenamiento, con el objetivo de minimizar la discrepancia entre las salidas predichas y las salidas reales.

2.2. Marco Conceptual

Inteligencia artificial: La inteligencia artificial (IA) es un campo de estudio de la informática que tiene como objetivo principal crear sistemas capaces de llevar a cabo actividades que generalmente requerirían de la inteligencia humana. (Ponce Cruz, 2010)

Alelo: En un algoritmo genético, un alelo se refiere a un valor específico que puede tomar un gen en una posición determinada del genoma. Los alelos representan las diferentes opciones o características que pueden tener los genes en una población.(Ponce Cruz, 2010)

Locus: En genética, un locus es una posición específica en un cromosoma donde se encuentra un gen o una secuencia de ADN particular. Cada locus puede contener uno o más alelos.(McCall, 2005)

Cromosoma: Un cromosoma en un algoritmo genético es una estructura que contiene un conjunto de genes o alelos que representan una posible solución. (Ponce Cruz, 2010)

Genotipo: El genotipo en un algoritmo genético se refiere a la configuración genética completa de un individuo o cromosoma. Representa la información genética contenida en el genoma, es decir, la combinación específica de genes y alelos que determina las características de la solución.(Ponce Cruz, 2010)

Fenotipo: El fenotipo en un algoritmo genético se refiere a la expresión o manifestación de las características genéticas en la solución. Es la forma en que el genotipo se traduce en un resultado observable o medible que representa la solución en el espacio del problema.(Ponce Cruz, 2010)

Exploración de regiones del espacio de búsqueda: La exploración de regiones del espacio de búsqueda es el proceso de buscar y examinar diferentes soluciones o puntos dentro de un espacio de búsqueda en busca de la solución óptima o deseada. En el contexto de algoritmos de optimización, implica evaluar diferentes posibilidades para encontrar la mejor opción.(Gutiérrez Reina et al., 2020)

Explotación de regiones de espacio de búsqueda: La explotación de regiones de espacio de búsqueda se refiere al proceso de utilizar información obtenida de soluciones previas o conocimiento adquirido para enfocar la búsqueda en áreas más prometedoras del espacio de búsqueda. Se busca aprovechar el conocimiento existente para mejorar la eficiencia y calidad de las soluciones encontradas.(Gutiérrez Reina et al., 2020, p. 43)

Restricción: En diversos campos, como la programación, la optimización o el diseño, las restricciones se utilizan para establecer límites o reglas que deben respetarse al buscar una solución o tomar decisiones. Las restricciones pueden ser de diferentes tipos, como restricciones numéricas, lógicas, de tiempo o de recursos.(García Serrano, 2017)

Hiperparámetro: Es un parámetro que se utiliza para controlar el comportamiento y rendimiento de un algoritmo de aprendizaje automático, pero que no es aprendido directamente a partir de los datos (Gorgolis et al., 2019).

2.3. Antecedentes empíricos de la investigación

Qi Li et al. (2023) en su artículo considero como aporte central de esta investigación presentar y desarrollar un enfoque mejorado que combina el Algoritmo Genético Simbólico (SGA) con la Red Neuronal de Memoria a Corto y Largo Plazo (LSTM) para anticipar los rendimientos de acciones en el mercado chino. Con el objetivo de hacer frente a los desafíos asociados a los indicadores fundamentales en mercados eficientes y a la baja frecuencia de actualizaciones de estos indicadores para modelos de aprendizaje profundo (DNN), se buscó evaluar la eficacia del marco propuesto. La metodología se fundamentó en el uso de datos que abarcaron indicadores fundamentales de 4,500 acciones chinas, implementando el SGA y LSTM junto con técnicas de aumento de datos y selección de características. Los resultados mostraron mejoras significativas en el Coeficiente de Información de Rango (Rank IC) y la Proporción de Información de IC (ICIR), con incrementos del 1,128% y 5,360%, respectivamente, para datos fundamentales. Además, la estrategia basada en el modelo SGA-LSTM superó a los índices bursátiles chinos, generando rendimientos anualizados en exceso notables en comparación con los índices CSI 300 y CSI 500. Este estudio resalta la eficacia del marco del algoritmo genético SGA- y la red neuronal LSTM mejorado en la predicción de rendimientos de acciones en el mercado chino superando los enfoques convencionales.

Mousapour et al. (2023) en su artículo presento como propósito principal desarrollar enfoques prácticos e innovadores para detectar señales falsas generadas por indicadores de análisis técnico en el mercado de metales preciosos. Estos indicadores, ampliamente utilizados en los mercados financieros, suelen presentar errores notables al identificar señales de compra y venta. A lo largo de un período de 10 meses, se llevó a cabo un análisis exhaustivo de estos indicadores utilizando registros de señales recopilados. Los resultados obtenidos indican que los metaheurísticos basados en redes neuronales propuestos son valiosos como herramienta de apoyo para la toma de decisiones de inversores, permitiéndoles abordar y gestionar las incertidumbres inherentes a los mercados financieros y de metales preciosos. Este estudio subraya la eficacia de los metaheurísticos propuestos, respaldados por la combinación de técnicas innovadoras, para mejorar la precisión de las señales de los indicadores de análisis técnico siendo en específico el caso de una red neuronal combinada con un algoritmo genético en donde se logró una precisión del 92%, ofreciendo

así una herramienta valiosa en entornos financieros caracterizados por considerables niveles de incertidumbre.

Bo (2023) en su artículo tuvo como objetivo optimizar la composición de las carteras, reducir riesgos y aumentar los rendimientos potenciales mediante el análisis de las interdependencias y correlaciones entre activos financieros mediante el uso de redes complejas, y al mismo tiempo empleando algoritmos genéticos como técnica de optimización. Los resultados evidencian que la cartera resultante de la aplicación de algoritmos genéticos a redes complejas muestra notables capacidades de control de riesgos considerando que este estudio tuvo un total de retorno de 26,67% y una ratio de Sharpe de 6,85% a diferencia de la cartera generada a partir de la teoría de Markowitz con un 10,58% de retorno y 2,99% de valor para el ratio de Sharpe. Este enfoque integrado minimiza de manera efectiva los riesgos asociados con las inversiones, contribuyendo así a la creación de una cartera más sólida y resistente, especialmente en entornos de mercados financieros volátiles.

Vasiani et al. (2020) en su artículo tuvo como objetivo optimizar las carteras de acciones mediante el uso del método del índice de prioridad y el algoritmo genético. Se pretendió reducir el riesgo de inversión y maximizar el rendimiento obtenido. Para lograr esto, se lleva a cabo una metodología que implica la selección de acciones basada en el índice de prioridad, el cual considera parámetros como el precio/ganancias (P/E), las ganancias por acción (EPS), la creación de riqueza, la subvaloración y el precio por ganancias/crecimiento (PEG). Se eligen las acciones de cada sector que obtengan una puntuación de índice de prioridad igual o superior a la puntuación mínima de las acciones seleccionadas. Además, se utilizó un algoritmo genético para determinar el porcentaje de inversión en cada acción seleccionada. Los resultados obtenidos revelan que aumentar el valor de los parámetros de escala no siempre conlleva un aumento en el rendimiento promedio de la cartera de acciones siendo el retorno del 1.65% en el escenario pesimista, 3.02% en el más probable y 4,71% en el optimista. Además, se destaca que la selección de acciones basada en datos diarios arroja un rendimiento promedio superior en comparación con el uso de datos anuales. Por último, se concluyó que el método presenta un período óptimo de hasta cinco meses para la toma de decisiones de inversión, por lo que la importancia de este artículo se tomó como referencia para evaluar el periodo que se tomarán para analizar los activos de capital y la estructura del algoritmo genético en los resultados de diferentes escenarios.

Lim et al. (2020) en su artículo formuló como objetivo principal diseñar una cartera óptima utilizando un Algoritmo Genético (AG), que incorpora la estrategia de impulso y las valoraciones de los activos. En términos de metodología, se lleva a cabo un análisis de los rendimientos ajustados por riesgo en períodos anteriores, utilizando la inercia de estos rendimientos como impulso para la selección de acciones. Sin embargo, se reconoce que los movimientos históricos por sí solos no son suficientes para predecir cambios futuros ni garantizar retornos positivos. Los resultados del estudio se basan en la prueba del modelo de AG en dos mercados diferentes, el S&P500 y el KOSPI200, en donde se proyectan ganancias superiores en comparación con el método anterior que solo utiliza la estrategia de impulso y los índices de mercado como referencia. Se concluyó que, se confirma que la combinación del Algoritmo Genético, la estrategia de impulso y las valoraciones de activos permite diseñar carteras óptimas en términos de rendimiento.

Para la presente investigación el aporte del artículo es de evaluar entornos adicionales como son las estrategias de impulso y valoración de activos, que podrían ser determinantes en la selección de individuos más aptos en el algoritmo.

Candia et al. (2020), en su artículo tuvo como objetivo abordar el problema de la selección de portafolio de proyectos para la adjudicación de interventorías de obra pública a través de concursos de méritos abiertos supervisados por el Instituto Nacional de Vías (INVIAS) en Colombia. Metodológicamente se evaluaron dos enfoques alternativos: un algoritmo genético meta-optimizado (GA) y un procedimiento de búsqueda voraz adaptativo probabilista meta-optimizado (GRASP). Para el caso de estudio de una Compañía con 207 contratos de trayectoria en el sector, se aplicaron ambas metaheurísticas y se compararon sus resultados en donde se encontró que ambas técnicas lograron encontrar puntajes de valoración óptimos en distintas instancias de prueba. Sin embargo, el algoritmo genético GA mostró un mejor desempeño de manera consistente en todas las evaluaciones realizadas. En algunos casos, el GA logró encontrar hasta 10 portafolios óptimos en menos de 9 minutos. En conclusión, se determinó que el algoritmo genético GA es más eficiente y efectivo para la selección de portafolio de proyectos en el contexto de los concursos de méritos abiertos supervisados por el INVIAS, por lo que la contribución principal se enfoca en probar dos tipologías distintas de algoritmos para obtener el mejor resultado en términos de eficiencia, lo cual encamina a la presente investigación a acotar los métodos usados.

Rodríguez et al. (2020) en su artículo mencionan que el objetivo de este trabajo fue proponer un método alternativo utilizando algoritmos evolutivos, específicamente un Algoritmo

Genético Canónico, para diseñar un portafolio de inversiones denominado "portafolio de divisas". Metodológicamente a nivel descriptivo se seleccionaron seis monedas en relación al peso mexicano: Guaraní Paraguayo, peso uruguayo, boliviano, dólar americano, libra esterlina y euro. El método utilizado tradicionalmente para construir un portafolio óptimo es la Teoría de Portafolio Óptimo de Harry Markowitz, que utiliza una serie de ecuaciones resueltas mediante el método de programación no lineal GRG. Los resultados demostraron que el Algoritmo Genético fue la mejor opción en términos de cálculo, superando al GRG. En conclusión, el uso de algoritmos evolutivos, en particular el Algoritmo Genético Canónico, se mostró como una alternativa eficiente para el diseño de un portafolio de inversiones en divisas. Considerando esto, sirve como referencia para tomar en cuenta en esta investigación que se pueden lograr soluciones más efectivas y adecuadas, especialmente en situaciones de incertidumbre económica y financiera mediante los fundamentos básicos de los algoritmos genéticos.

Maholi et al. (2019) en su artículo tuvo como objetivo principal utilizar redes neuronales artificiales (ANN) para predecir los valores futuros de las acciones y, a continuación, emplear un algoritmo genético (GA) para formar carteras óptimas que maximicen el rendimiento y minimicen el riesgo. En cuanto a la metodología utilizada, se comienza aplicando la ANN, un modelo de aprendizaje automático inspirado en la actividad de las células cerebrales humanas, para realizar predicciones de los valores futuros de las acciones. Los resultados obtenidos en este estudio indican que la implementación del GA como alternativa al método SIM logra una mejor optimización en términos de rendimiento. La combinación de la ANN para realizar predicciones precisas y el GA para formar carteras óptimas demuestra ser eficaz en la maximización del rendimiento y la minimización del riesgo en comparación con enfoques tradicionales. En conclusión, este estudio demuestra que el uso de redes neuronales artificiales (ANN) en combinación con algoritmos genéticos (GA) permite mejorar la optimización de las carteras de acciones, y de la misma forma muestra una eficiencia mayor a nivel computacional al tener un error de 0.017 y un precisión de 0.9814 en su topología óptima por lo que el aporte de este artículo se fundamenta en evaluar el uso de más de una tecnología de inteligencia artificial para optimizar los portafolios de inversión.

Chen et al. (2019) en su artículo se enfocó en la optimización de carteras de acciones en grupo, debido a la naturaleza atractiva de los conjuntos de datos en los mercados financieros. Esta investigación ofrece una metodología para la formación de carteras de acciones en

grupo que considera la diversidad y los aspectos financieros relevantes para maximizar las ganancias. Como resultados se ha demostrado que el uso de la función de aptitud f_2 , que considera los precios de las acciones y las unidades compradas, mejora el equilibrio de precios y unidades en la cartera, aunque puede afectar la satisfacción general de la cartera. En general, los resultados indican que el enfoque propuesto ha logrado una mayor diversidad y equilibrio en las carteras de acciones en grupo.

Este estudio refuerza el concepto de diversificación usado en la presente investigación al mejorar la selección de agrupamiento de los activos de capital.

Liagkouras, K. (2019) en su artículo tuvo como objetivo abordar las limitaciones de las técnicas de algoritmos evolutivos existentes en la resolución de problemas combinatorios a gran escala, debido a su amplio espacio de búsqueda, lo cual dificulta el examen de instancias del mundo real de gran tamaño. La metodología consiste en probar el rendimiento del algoritmo propuesto en la asignación óptima de recursos limitados a un número de oportunidades de inversión competitivas para optimizar los objetivos. Los resultados obtenidos muestran que el algoritmo propuesto es altamente eficiente en términos de tiempo de procesamiento, donde se demuestra que un grupo de 100 individuos en la población es suficiente para manejar de manera eficiente las instancias de prueba mencionadas previamente. En consecuencia, el algoritmo propuesto TDMEA puede resolver eficientemente problemas de gran escala utilizando tamaños de población relativamente moderados. En conclusión, se ha desarrollado un nuevo algoritmo que supera las limitaciones de los enfoques existentes en la resolución de problemas combinatorios a gran escala.

Esta investigación sirve como base para evaluar los distintos tipos de algoritmos para el análisis de Pareto, en el caso comparativo de la evaluación de los dos parámetros principales, de rendimiento y volatilidad.

2.4 Hipótesis

2.4.1. Hipótesis General

El diseño de la red neuronal artificial y el algoritmo genético optimizan el portafolio de inversión

Esta hipótesis se encuentra argumentada debido a que en investigaciones como la de Qi Li et al. (2023) se usaron redes neuronales LSTM combinadas con algoritmos genéticos,

para este caso en específico evaluaron 4,500 acciones chinas y alcanzo un rendimiento promedio entre 1,128% y 5,360%, de la misma forma, Mousapour et al. (2023) usaron una red neuronal hibrida bajo el enfoque de metaheurísticas de predicción alcanzado al utilizar estas tecnologías una precisión del 92%. Otros autores como Bo (2023) usó redes neuronales complejas combinadas con algoritmos genéticos en donde el retorno alcanzo un 0.2667% y por último Maholi et al. (2019) desarrollaron un algoritmo de redes neuronales densas donde se demostró una eficiencia mayor a nivel computacional al tener un error de 0.017 y una precisión de 0.9814 obteniendo un 0.28% de retorno y riesgo de 1.86%.

2.4.2. Hipótesis específicas

- El diseño de la red neuronal artificial permite predecir la tendencia de los activos de capital
- La función objetivo, representación del cromosoma de los individuos, e hiperparámetros mejoran la calidad del algoritmo genético
- Al evaluar las diversas configuraciones del algoritmo genético se determina un desempeño eficiente representado en un valor de fitness alto

2.5 Identificación de variables e indicadores

Las variables son el algoritmo genético y la red neuronal artificial, considerando que es el elemento dentro de la investigación al cual se le definirá su arquitectura a nivel de diseño, esto a partir de la optimización del portafolio de inversiones.

2.6. Operacionalización de variables

Variables	Definición conceptual	Definición operacional	Dimensiones	Indicadores	Técnicas e instrumentos
Red Neuronal Artificial	También conocida como célula o autómeta, es el elemento que contiene un estado en su interior, el cual se denomina nivel o grado de activación, y que a partir de señales que la estimulan puede cambiar su estado (Isasi & Galván, 2004).	Modelo de inteligencia artificial cuya finalidad es generar de forma precisa la predicción de los valores de las acciones de las empresas evaluadas, y su medida de rendimiento se considera la raíz del error cuadrático medio y el error absoluto medio	Arquitectura de la red	Precisión del modelo	Análisis exploratorio
			Desempeño del modelo	Error de pronóstico	Preprocesamiento de Datos
Algoritmo Genético	Los algoritmos genéticos son algoritmos de búsqueda basados en mecanismos de la selección natural y genética natural (Goldberg, 1989).	Modelo de inteligencia artificial que procesa y genera un portafolio de inversión optimo mediante la exploración y explotación de soluciones en regiones de individuos factibles dentro del universo muestral.	Diseño del cromosoma y del fitness	Valor del fitness	Bases de datos de la Bolsa de Valores de Lima
			Tamaño de población y generaciones	Rendimiento del portafolio	Análisis estadístico
				Volatilidad del portafolio	Visualización de datos
					Ajuste de hiperparámetros
					Desarrollo algorítmico
					Validación del modelo

CAPITULO III: METODOLOGÍA

3.1. Ámbito de estudio

Al realizar la evaluación de activos de capital de la Bolsa de Valores de Lima [BVL], el alcance del estudio a nivel geográfico abarca el Perú, considerándose que los datos de entrada que son valores cuyo entorno en el que se definen está dentro del ámbito nacional.

3.2. Tipo y nivel de investigación

El tipo de investigación es aplicada, debido a que, a partir de la teoría básica o fundamental, como lo es la teoría de portafolios de inversión, así como la teoría de algoritmos genéticos, se aplica a un caso práctico tal como menciona Arias (2021). Para el caso de la presente investigación se aplica en un entorno real como es la conformación de una cartera de inversión a partir de los activos de capital que cotizan en la Bolsa de Valores de Lima.

El nivel de investigación es explicativo, considerando que se pretende especificar por qué ocurre un evento y en qué condiciones se manifiesta, tal como mencionan Hernández & Mendoza (2018), para el caso de la presente investigación, al diseñar el algoritmo genético se debe experimentar con los hiperparámetros ajustando sus valores para encontrar cuales logran optimizar el portafolio de inversión, y de la misma forma predecir el valor óptimo del portafolio en cuanto a rendimiento y volatilidad.

3.3. Unidad de análisis

La unidad de análisis son los activos de capital en el mercado de valores nacional, que cotizan en la Bolsa de Valores de Lima

3.4. Población de estudio

La población de estudio son los activos del mercado de capitales nacional, los cuales son una cantidad de 255 en el periodo de desarrollo de la investigación

3.5. Tamaño de muestra

Para la determinación de la muestra se evaluó a toda la población, es decir a las 255 empresas que conforman parte de la Bolsa de valores y cuyo análisis comenzó explorando los datos respecto a la renta variable que presentaron.

3.6. Técnicas de selección de muestra

La técnica de la selección de la muestra es determinística por conveniencia, teniendo en cuenta que existe la disponibilidad de datos para evaluar a toda la población, que fueron todas las empresas que cotizan en la Bolsa de Valores de Lima.

3.7. Técnicas de recolección de información

Los inputs para el modelamiento del algoritmo genético y de la red neuronal, fueron datos históricos de la Bolsa de Valores de Lima, por lo que su recolección fue por medio de fuentes secundarias adquiridos directamente de la web oficial de la entidad mencionada.

3.8. Técnicas de análisis e interpretación de la información

Para la primera fase de procesamiento de los datos de entrada, se realizó un análisis de tendencia por medio del diseño de una red neuronal artificial, para esto, los datos de los activos de capital fueron sometidos a un análisis de independencia, con la finalidad de cumplir con el concepto de diversidad en la cartera y que estos no tengan una correlación que pueda sesgar el análisis; luego de ser procesados por la red neuronal artificial se dio como resultado el conocer cuáles activos de capital tienen una tendencia fuerte a tener valores positivos en los siguientes valores temporales. De forma consecuente al identificar los activos de capital con mayor capacidad de rendimientos positivos se procedió a modelar el diseño del algoritmo genético mediante la definición cromosómica de los individuos que representaron las soluciones al problema, así como también el modelamiento de la función objetivo. Luego se procedió a definir el tipo de operador genético a usar, es decir, determinar qué tipo de selección, cruzamiento y mutación uso en el algoritmo, así como sus hiperparámetros respectivos; y finalmente se analizó la estabilidad de los diseños.

Cabe señalar que todo el procesamiento de datos, el diseño de la red neuronal y desarrollo del algoritmo genético, fueron codificados en lenguaje de programación Python mediante el entorno de desarrollo Spyder, y las respectivas librerías para procesamiento de datos, Tensor Flow, Keras, Deap, Scipy, y matplotlib.

3.9. Técnicas para demostrar la verdad o falsedad de las hipótesis planteadas

Respecto a las carteras de inversión óptimas estas se sometieron a su respectivo análisis comparativo y determinación del mejor diseño, mediante la evaluación de sus indicadores de rendimiento y riesgo.

CAPITULO IV: RESULTADOS Y DISCUSIÓN

4.1. Resultados

4.1.1. Red Neuronal artificial para predecir la tendencia de los activos de capital

Para el primer objetivo se planteó diseñar una red neuronal artificial con una topología cuya finalidad sea predecir la tendencia los activos de capital, y con esto obtener los inputs para al algoritmo genético, que son un conjunto de cotizaciones de los activos de capital que tengan una tendencia positiva, asegurando que la cartera de inversión estará conformada por elementos que garanticen rendimientos positivos y superiores al periodo anterior.

Dentro del análisis para poder definir las entradas de la red neuronal se encontró que la información disponible debe tener claramente definido su origen, en este caso a partir de los datos públicos de la Bolsa de valores, tal como consideran los autores de Greiff & Rivera (2018) respecto a la bolsa de valores de Colombia, Maholi et al. (2019) con las acciones de la bolsa de valores de Indonesia, y Conti et al. (2005) para la bolsa de valores de Venezuela. Para el caso de la presente investigación es la Bolsa de Valores de Lima, es decir la bolsa en la que cotizan las acciones que serán evaluadas en las que se contempla para cada activo de capital el precio de apertura es decir el precio en que inicia a cotizar el activo, el precio de cierre con el que finaliza su valor en el día, el máximo valor de cotización durante el día, el mínimo valor que cotizo el activo durante el día, el precio promedio en el que cotizo la acción, la cantidad monetaria negociada respecto al activo durante el día, y el valor del activo de cierre anterior corregido.

A partir de esa fuente de información se planteó el desarrollo de las topologías específicas de las redes neuronales a evaluar para encontrar la más eficiente respecto a la predicción de la tendencia, y su utilidad estuvo enfocada en identificar aquellas acciones que tienen una tendencia positiva para asegurar rendimientos positivos y que sirvieron como entradas al algoritmo genético. Para el caso específico, se usó de topologías de tipo de red neuronal recurrente llamadas redes Long-short Time Machine [LSTM] cuyas características se ajustan al tipo de datos que proporciona la Bolsa de valores de Lima cuando se trata de activos de capital como son las series de tiempo.

Análisis exploratorio para datos de la Bolsa de Valores de Lima: Para obtener entradas al modelo de red neuronal, se desarrolló un pretratamiento de los datos, con la finalidad de que estos se estandaricen y sean de utilidad para que la red genere datos consistentes en su proceso de aprendizaje y uso general; por tanto el proceso se orienta a partir del procedimiento descrito por Ruochen & Muchao (2021) en donde inicia por el manejo de datos faltantes, esta es una característica fundamental para que sean entradas de la red neuronal. Adicional a lo antes descrito, para el caso de los datos de la Bolsa de Valores de Lima, se necesitó un tratamiento de datos adicional antes de llegar al proceso mencionado, debido a que los datos deben estar uniformemente descritos, y estandarizados temporalmente en todos los activos evaluados.

Para el desarrollo de la obtención de datos se hizo en un solo corte temporal en donde de forma escalonada se fue descartando los activos de acuerdo a los datos que iban presentando y filtrando; inicialmente en el portal de la BVL presenta el registro y datos de 255 empresas, de estas empresas solo el 85% (216) presentaban presentan información de renta variable. Al explorar los archivos CSV solo el 57% del total (145) de empresas tenían información de data histórica, el resto solo presento la fecha en que se debió registrar los parámetros de comportamiento del mercado; luego se procedió a encontrar el rango temporal mayor de los datos, es decir el límite inferior y superior, encontrándose que las empresas mostraron datos desde el 2 de enero del 2012 fueron 38.43% (98) empresas, y el 18% (46) presentaron los datos de cotización a partir de años anteriores, por lo que se definió como rango inferior la fecha antes mencionada. En cuanto al rango superior, debido a que el año de desarrollo de la investigación aun no culmina se consideró tomar el límite superior el 2 de enero del 2023, para que el rango tenga simetría en cuanto a periodos completos. Luego se procedió a evaluar cada uno de los registros, encontrándose que el rango mayor de datos que presentaron 87% (85) las empresas filtradas en esta etapa fueron de 2764 registros, y solo 13% de las empresas presentaron fechas faltantes. Por lo que se procedió a evaluar las empresas que presentaron esta cantidad de registros, en donde en el análisis se encontró que había datos de los parámetros de los activos faltantes, por lo que se procedió a evaluar el porcentaje de datos que estaban ausentes, con la finalidad de encontrar una tolerancia que no afecte el comportamiento del modelo a diseñar.

Según la tabla 1, dependiendo de los 9 parámetros que exponen el comportamiento de las acciones los cuales fueron, precio de apertura, precio de cierre, precio máximo, precio mínimo, precio promedio, cantidad negociada, monto negociado, fecha anterior, cierre

anterior corregido, se agrupo en 10 grupos que representan la cantidad de datos faltantes, esto para poder conocer el número de datos faltantes en el total de las 86 empresas evaluadas y guardando la congruencia respecto a la cantidad de datos faltantes que según autores como Dagnino (2014) y Bennett (2001) recomiendan un margen de tolerancia admisible es de 10% considerando que la ausencia de un porcentaje mayor puede generar sesgos o la pérdida de la distribución natural de los datos; esto representado en la tabla 1 en donde se exponen 10 rangos, en donde cada uno representan un intervalo de 10% de datos faltantes, es decir en R₁ estarán las empresas que tienen entre 0 a 10% de datos faltantes, R₂ contendrá a las empresas que tienen entre 10 a 20% de datos faltantes, y así sucesivamente.

Tabla 1
Representación porcentual de empresas según datos faltantes

Rango de datos	Porcentaje de empresas según la cantidad de datos faltantes								
	Apertura	Cierre	Máximo	Mínimo	Promedio	Cantidad negociada	Monto negociado (S/)	Fecha anterior	Cierre anterior corregido
R ₁	8%	8%	8%	8%	12%	12%	12%	58%	59%
R ₂	6%	6%	6%	6%	4%	4%	4%	19%	19%
R ₃	4%	4%	4%	4%	6%	6%	6%	9%	11%
R ₄	4%	4%	4%	4%	1%	1%	1%	5%	4%
R ₅	0%	0%	0%	0%	5%	5%	5%	6%	6%
R ₆	6%	6%	6%	6%	5%	5%	5%	0%	0%
R ₇	8%	8%	8%	8%	6%	6%	6%	0%	0%
R ₈	8%	8%	8%	8%	11%	11%	11%	1%	1%
R ₉	13%	13%	13%	13%	11%	11%	11%	1%	0%
R ₁₀	45%	45%	45%	45%	42%	42%	42%	2%	2%

Elaboración: Propia

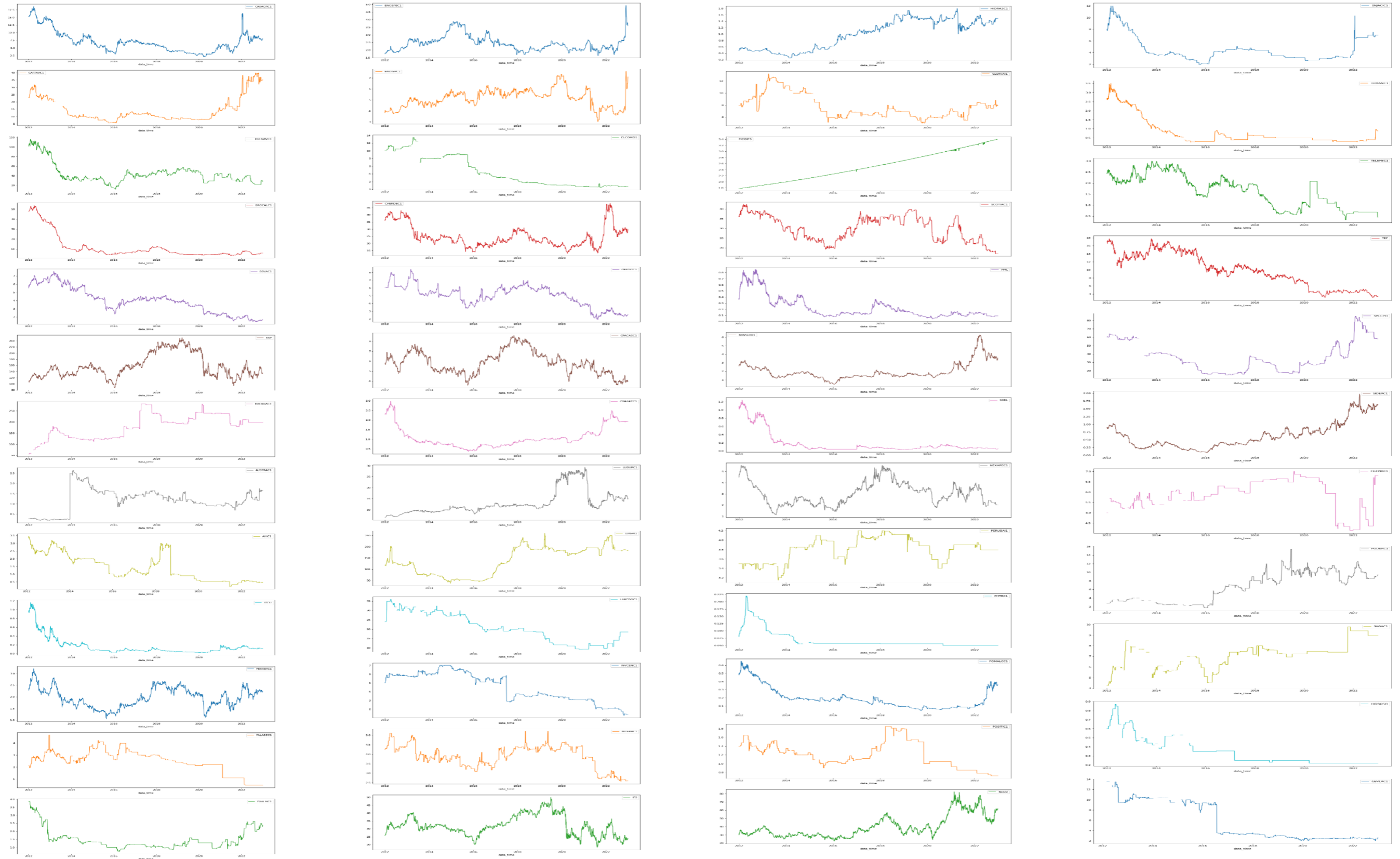
Como se puede observar solo el parámetro de cierre anterior corregido cumple con un porcentaje representativo del 59% que representan 50 empresas en el primer intervalo, cuyos datos faltantes son menores al 10%, mientras que para los demás parámetros en este rango, ninguno cumple con el número de empresas necesarias, definido como cantidad mínima de 9 empresas para el desarrollo del algoritmo, por lo que de esta manera se determina que el parámetro que servirá para generar la proyección de la red neuronal será el cierre anterior corregido, y que define el uso de una red neuronal recurrente LSTM modelo univariado multi-step.

Al haber determinado los activos de capital con los que se trabajó se procedió a identificar algún outliers en los activos de capital, así como también completar los datos

faltantes, ya que es un requisito para que el algoritmo pueda generar resultados consistentes, considerando que los datos deben estar uniformemente descritos. Para esto se hizo una revisión grafica del comportamiento de los datos haciendo uso de la librería matplotlib del lenguaje de programación Python, mostrándose para los activos de capital como se indica en la figura N° 24:

Figura 24

Representación gráfica de las acciones de las empresas antes del tratamiento de outliers y datos faltantes



Como se puede observar en el conjuntos de gráficos, no se encontraron outliers, sin embargo cabe mencionar que para el caso de los activos identificados con los nemónicos ECASAGRC1, SNJACIC1 existen dos puntos pronunciados en la gráfica sin embargo son menores a sus máximos históricos por lo que no se consideran datos anómalos, para el caso ENGEPEC1 se puede identificar un pico pronunciado, sin embargo antes y después de este dato existen datos que acompañan este comportamiento y se hace evidente en la serie de tiempo un comportamiento de mayor valorización del activo en ese periodo. También dentro de las gráficas se puede observar datos faltantes para 24 empresas en las series de tiempo. Las cantidades específicas de datos faltantes para cada una de las 24 empresa se muestran en la tabla 2:

Tabla 2

Datos faltantes del registro de acciones de las empresas

Nemónico	Cantidad de datos faltantes
CARTAVC1	91
BUENAVC1	14
BACKUAC1	92
AIHC1	25
FALABEC1	24
EXALMC1	59
ELCOMEI1	64
CREDITC1	2
LAREDOC1	81
HIDRA2C1	6
GLORIAI1	26
PERUBAI1	16
PHTBC1	81
POSITIC1	99
SNJACIC1	37
TUMANC1	9
TEF	9
SPCCPI1	85
SIDERC1	1
EGEPIBC1	248
PODERC1	283
SAGAC1	192
HIDROSI1	215
GBVLAC1	228

Estos datos se tratarán con métodos específicos de imputación de interpolación, de mediana y de K-means a continuación:

Imputación por interpolación: primer método se usó la interpolación en sus 3 versiones, lineal, cuadrático y cubico, tal como se muestra a continuación:

$$f(x) = y_0 + \frac{(x-x_0)(y_1-y_0)}{x_1-x_0} \dots \dots \dots \text{Ecuación lineal}$$

$$f(x) = y_0 \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} + y_1 \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} + y_2 \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} \dots \dots \dots \text{Ecuación cuadrática}$$

$$f(x) = a + b(x - x_i) + c(x - x_i)^2 + d(x - x_i)^3 \dots \dots \dots \text{Ecuación cubica}$$

Donde:

$$a = y_i$$

$$b = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}$$

$$c = \frac{3(y_{i+2} - y_{i+1}) - b(x_{i+2} - x_{i+1}) - 2b(x_{i+1} - x_i)}{(x_{i+2} - x_{i+1})^2}$$

$$d = \frac{2(b(x_{i+1} - x_i) + b(x_{i+2} - x_{i+1}) - 3(y_{i+1} - y_i))}{(x_{i+2} - x_{i+1})^3}$$

$\forall (x_i, y_i)$ que son pares ordenados donde $i = 1, 2, 3, \dots, n$

cabe resaltar que la evaluación de la imputación será por una comparación de medias antes y después de la imputación, encontrándose como resultados para el método de interpolación, lo mostrado en la siguiente tabla3:

Tabla 3
Variaciones porcentuales del precio de cierre después de la imputación polinómica

Nemónico	Medias originales	Interpolación		Porcentaje de variación
		Lineal	Cuadrática/Cubica	
CARTAVC1	S/ 14.52	S/	14.38	0.96%
BUENAVC1	S/ 44.05	S/	44.11	0.14%
BACKUAC1	S/ 178.20	S/	180.71	1.41%
AIHC1	S/ 1.42	S/	1.40	1.41%
FALABEC1	S/ 2.49	S/	2.48	0.40%
EXALMC1	S/ 1.49	S/	1.49	0.00%
ELCOMEI1	S/ 4.56	S/	4.44	2.63%
CREDITC1	S/ 5.01	S/	5.01	0.00%
LAREDOC1	S/ 20.65	S/	20.34	1.50%
HIDRA2C1	S/ 0.99	S/	0.99	0.00%
GLORIAI1	S/ 7.73	S/	7.71	0.26%
PERUBAI1	S/ 3.81	S/	3.81	0.00%
PHTBC1	S/ 0.07	S/	0.07	0.00%

POSITIC1	S/	1.21	S/	1.20	0.83%
SNJACIC1	S/	4.51	S/	4.52	0.22%
TUMANC1	S/	0.80	S/	0.80	0.00%
TEF	S/	9.84	S/	9.83	0.10%
SPCCPI1	S/	36.69	S/	36.36	0.90%
SIDERC1	S/	0.63	S/	0.63	0.00%
EGEPIBC1	S/	5.83	S/	5.86	0.51%
PODERC1	S/	6.51	S/	6.90	5.99%
SAGAC1	S/	7.12	S/	7.14	0.28%
HIDROSI1	S/	0.35	S/	0.34	2.86%
GBVLAC1	S/	5.90	S/	5.42	8.14%
porcentaje de empresas sin variación en sus medias después de la imputación				29.2%	

Como se puede observar solo un 29.2% de empresas no sufren cambios respecto a su media al utilizar las variantes de interpolación en sus tres versiones, esto se refleja en la columna de porcentaje de variación en donde 7 empresas presentaron un valor igual a 0%.

Imputación por mediana: Para el segundo caso se usó el método de la imputación por medianas, el cual es un proceso análogo como el recomendado por Pratama et al. (2016) en donde se utiliza las medias y las modas, sin embargo en una serie de tiempo estas medidas estadísticas tienen a sesgar el comportamiento de la variable, por lo que se optó por usar la mediana considerando que al ser una serie de tiempo, el dato que suaviza la tendencia en un rango específico será el valor central de la serie, por lo que la tabla 4 muestra los siguientes resultados:

Tabla 4

Variaciones Porcentuales después de la imputación por medianas

Nemónico	Imputación			
	Medias originales	por mediana	Porcentaje de variación	
CARTAVC1	S/ 14.38	S/ 14.25	0.90%	
BUENAVC1	S/ 44.11	S/ 44.09	0.05%	
BACKUAC1	S/ 180.71	S/ 180.92	0.12%	
AIHC1	S/ 1.40	S/ 1.40	0.00%	
FALABEC1	S/ 2.48	S/ 2.49	0.40%	
EXALMC1	S/ 1.49	S/ 1.48	0.67%	
ELCOMEI1	S/ 4.44	S/ 4.41	0.68%	
CREDITC1	S/ 5.01	S/ 5.01	0.00%	
LAREDOC1	S/ 20.34	S/ 20.32	0.10%	
HIDRA2C1	S/ 0.99	S/ 0.99	0.00%	
GLORIAI1	S/ 7.71	S/ 7.71	0.00%	
PERUBAI1	S/ 3.81	S/ 3.81	0.00%	
PHTBC1	S/ 0.07	S/ 0.07	0.00%	

POSITIC1	S/	1.20	S/	1.20	0.00%
SNJACIC1	S/	4.52	S/	4.52	0.00%
TUMANC1	S/	0.80	S/	0.80	0.00%
TEF	S/	9.83	S/	9.83	0.00%
SPCCPI1	S/	36.36	S/	36.25	0.30%
SIDERC1	S/	0.63	S/	0.63	0.00%
EGEPIBC1	S/	5.86	S/	5.87	0.17%
PODERC1	S/	6.90	S/	6.96	0.87%
SAGAC1	S/	7.14	S/	7.15	0.14%
HIDROSI1	S/	0.34	S/	0.33	2.94%
GBVLAC1	S/	5.42	S/	5.25	3.14%
Porcentaje de empresas sin variación en sus medias después de la imputación				45.8%	

Como se puede observar, para este método el porcentaje de empresas que mantiene su media después de la imputación aumento en un 45.8%, siendo mayor al encontrado en la interpolación.

Imputación por k-means: Finalmente se usó el método de K-means, el cual es un algoritmo de machine learning que, para el caso específico de la imputación, genera un dato a partir de los k- datos vecinos al dato faltante tal como se recomienda en la revisión de Fang & Wang (2020) . El método se describe según lo mencionado por Reyes Osorio (2023) :

$$d(x_a, x_b) = \sqrt{\sum_{j=1}^n d_j(x_{aj}, x_{bj})^2}$$

donde $d(x_a, x_b)$ es la distancia que existe en los puntos x_a y x_b para el elemento j, es decir la distancia con los k vecinos más cercanos. Así mismo la ponderación que se le asigna a cada vecino está dada por:

$$d_j(x_{aj}, x_{bj}) = \begin{cases} 1 & \text{Si } (1 - m_{aj})(1 - m_{bj}) = 0 \\ d_N(x_{aj}, x_{bj}) & \text{Para cualquier variable cuantitativa} \end{cases}$$

Donde m es un vector con valores binarios que toma valores de 1 si el valor se desconoce y 0 si el valor se conoce, para el segundo factor conocido como la distancia de Manhattan se define:

$$d_N(x_{aj}, x_{bj}) = \frac{|x_{ai} - x_{bi}|}{\max(x_j) - \min(x_j)}$$

donde se obtiene los siguientes resultados mostrado en la tabla 5:

Tabla 5

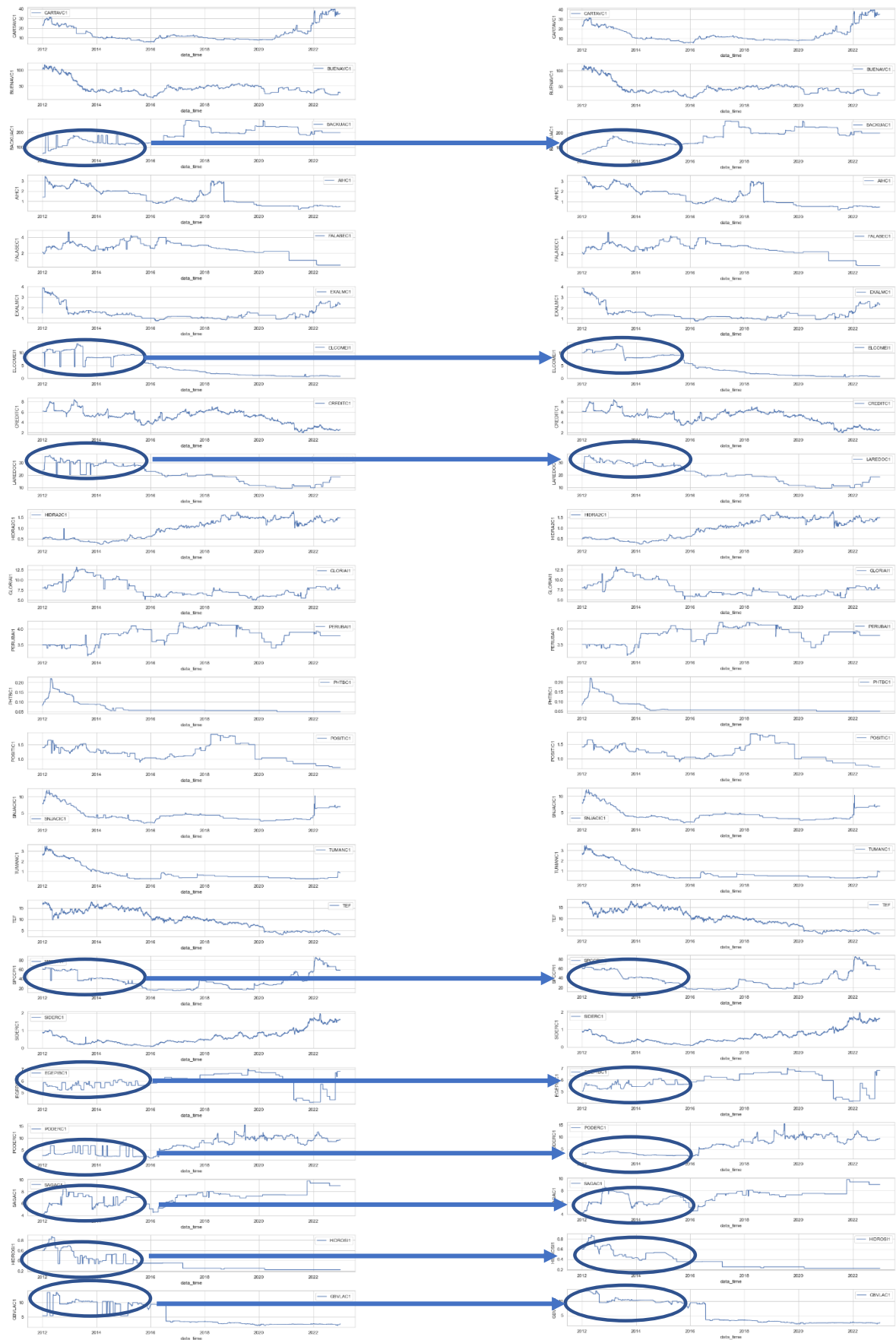
Imputación por el método KNN

Nemónico	Medias originales	Imputación KNN	Poncentaje de variación
CARTAVC1	S/ 14.38	S/ 14.38	0.00%
BUENAVC1	S/ 44.11	S/ 44.11	0.00%
BACKUAC1	S/ 180.71	S/ 180.71	0.00%
AIHC1	S/ 1.40	S/ 1.40	0.00%
FALABEC1	S/ 2.48	S/ 2.48	0.00%
EXALMC1	S/ 1.49	S/ 1.49	0.00%
ELCOMEI1	S/ 4.44	S/ 4.44	0.00%
CREDITC1	S/ 5.01	S/ 5.01	0.00%
LAREDOC1	S/ 20.34	S/ 20.34	0.00%
HIDRA2C1	S/ 0.99	S/ 0.99	0.00%
GLORIAI1	S/ 7.71	S/ 7.71	0.00%
PERUBAI1	S/ 3.81	S/ 3.81	0.00%
PHTBC1	S/ 0.07	S/ 0.07	0.00%
POSITIC1	S/ 1.20	S/ 1.20	0.00%
SNJACIC1	S/ 4.52	S/ 4.52	0.00%
TUMANC1	S/ 0.80	S/ 0.80	0.00%
TEF	S/ 9.83	S/ 9.83	0.00%
SPCCPI1	S/ 36.36	S/ 36.36	0.00%
SIDERC1	S/ 0.63	S/ 0.63	0.00%
EGEPIBC1	S/ 5.86	S/ 5.86	0.00%
PODERC1	S/ 6.90	S/ 6.90	0.00%
SAGAC1	S/ 7.14	S/ 7.14	0.00%
HIDROSI1	S/ 0.34	S/ 0.34	0.00%
GBVLAC1	S/ 5.42	S/ 5.42	0.00%
Porcentaje de empresas sin variación después de la imputación KNN		100.0%	

Como se puede observar para este método se obtuvo que el 100% de los datos mantiene su media después de la imputación, sin embargo, la revisión por medio de la figura 25 expresa una visión más general del comportamiento de los datos de los modelos:

Figura 25

Representación de comportamiento de los datos imputados



De las dos graficas se puede observar que el método KNN genera datos que se representan por picos pronunciados a diferencia del método de interpolación en donde se puede observar una representación gráfica más suavizada, por lo que se consideró conocer el tipo de método de imputación a usar teniendo como parámetro de referencia el coeficiente de determinación R^2 a partir de la generación de un modelo de tendencia lineal, obteniéndose los siguientes resultados mostrados en la tabla 6:

Tabla 6

Comparación de coeficientes de determinación de los métodos de imputación de datos

Nemónico	Modelo KNN	Interpolación
CARTAVC1	0.0293	0.0219
BUENAVC1	0.2269	0.2241
BACKUAC1	0.4843	0.5402
AIHC1	0.6604	0.6868
FALABEC1	0.5235	0.5271
EXALMC1	0.0598	0.0601
ELCOMEI1	0.8295	0.8646
CREDITC1	0.462	0.4627
LAREDOC1	0.8067	0.858
HIDRA2C1	0.7759	0.7802
GLORIAI1	0.3341	0.341
PERUBAI1	0.1559	0.1594
PHTBC1	0.4889	0.474
POSITIC1	0.1931	0.1992
SNJACIC1	0.1185	0.1139
TUMANC1	0.4978	0.4959
TEF	0.8633	0.869
SPCCPI1	0.0015	0.0001
SIDERC1	0.4874	0.4878
EGEPIBC1	0.0015	0.0002
PODERC1	0.6065	0.7313
SAGAC1	0.4295	0.4173
HIDROSI1	0.6927	0.7642
GBVLAC1	0.6884	0.8218
Porcentaje de representación de empresas con un coeficiente de determinación mayor	33%	67%

Teniendo en cuenta que en el modelo de interpolación un 67% de las empresas muestran un coeficiente de determinación mayor, se eligió este modelo como método de imputación.

Diseño de Red neuronal para la predicción de las acciones de entrada para el algoritmo genético: Para el desarrollo de este apartado, se diseñó una red neuronal recurrente, que por sus características de utilizar un estado anterior y poder generar el estado siguiente, se ajustan a los modelos de series de tiempo en donde han demostrado su utilidad en múltiples tareas de inteligencia artificial (Hewamalage et al., 2021). En el caso de la Bolsa de Valores de Lima la herramienta usada son las medias móviles mediante el análisis técnico, según se puede confirmar entrando al portal oficial de esta entidad. En el caso del presente estudio se planteó el diseño a partir del tipo de redes neuronales recurrentes denominadas Long Short-Term Memory [LSTM] las cuales tiene la capacidad de recordar a partir de compuertas de pérdida de memoria las cuales por medio de funciones no lineales eliminan o preservan información, compuertas de entrada las cuales deciden qué información es aceptada por el modelo y las compuertas de salida en donde se decide que parte de la memoria de la red LSTM contribuya a la salida.(Siarni-Namini et al., 2019)

La estructura de una red LSTM para este caso, considera el valor de cierre corregido de la serie temporal de valorización de las acciones tal como se mencionó en el análisis exploratorio; y su salida será una medida de tendencia definida bajo el concepto que la tendencia es alcista, si es que su valor de cotización es mayor al periodo anterior, sin embargo, esta expresión se debe complementar con una medida de holgura que mejore la precisión del límite que se puede aceptar como un valor de una tendencia alcista, a partir de los datos analizados, por lo cual se determinó que la expresión se complementaría tal como se expresa a continuación:

$$P_{(t+1)} > P_{(t)} + [\sigma_{\%Mercado}] * P_{(t)}$$

Donde:

- $P_{(t)}$: Precio de cierre de la acción en el periodo t
- $P_{(t+21)}$: Precio de cierre de la acción en el periodo $t + 21$
- $\sigma_{\%Mercado}$: Representación porcentual de la desviación estandar del mercado

Según la expresión anterior se usa la medida de desviación estándar del mercado para que la tendencia tenga un excedente que absorba cualquier variación que exista por alguna variable atípica dentro del mercado internacional, considerando que el 20% al 50% del comportamiento de los precios se generan a partir de las fuerzas del mercado (Gitman &

Joehnk, 2009, p. 357), por lo que insertar en la ecuación una medida que considere las fluctuaciones que viene teniendo el comportamiento general de las acciones, ayuda a mitigar cualquier condición que no asegure un rendimiento mayor en una temporalidad siguiente, la cual también es definida después del día 21 como la rentabilidad esperada en el corto plazo (Cáceres, 2018). Así mismo respecto a que indicador se usó para determinar esta medida de desviación es el Standard & Poor's 500 [S&P500] el cual es definido como un índice autentico, al considerar la capitalización de las 500 empresas más grandes de Estados Unidos, y captura el 80% de toda la capitalización.

Algoritmo Univariado-Multistep: Por lo formulado anteriormente se planteó 4 topologías de redes LSTM para su evaluación, en donde se consideran como hiperparámetros la entrada de la Red que serán lotes de 42 días, y una salida de 21 días, la cual es el valor de cierre de la acción de cada empresa, así mismo el número de neuronas en cada capa de la red para cada topología, el número de épocas y la ratio de aprendizaje, bajo la siguiente arquitectura mostrada en pseudocódigo en la tabla 7:

Tabla 7

Pseudocódigo de la red neuronal artificial

Algorithm 1 Modelo de Red LSTM Univariado-Multistep

```

1: Entrada: Librerías
2: Entrada: Acciones del análisis exploratorio
3: Entrada: hiperparametros
4: Salida: Métricas de rendimiento del algoritmo
5: Data frame ← Valores de las acciones
6: tr, vl, tst ← set de entrenamiento, validación y test
7: function ESCALAMIENTO DE DATOS
8:   Set de entrenamiento escalado ← Escalador MinMax Scaler(Set de entrenamiento)
9:   Set de validación escalado ← MinMax Scaler(Set de validación)
10:  Set de prueba escalado ← MinMax Scaler(Set de prueba)
11: fin function
12: Para empresas del data frame hacer
13:   function ENTRENAR MODELO
14:     Modelo ← Secuencial
15:     Modelo ← agregar capa LSTM
16:     Modelo ← agregar capa Densa
17:   fin function
18:   function METRICAS DE RENDIMIENTO
19:     RMSE, MAE Precisión ← Desempeño del modelo
20:   fin function
21: fin Para
22: Salida: Métricas del rendimiento del modelo

```

El algoritmo resumido en pseudocódigo contiene en su estructura, bloques de entrada y de salida, así como las funciones de entrenamiento, las métricas de rendimiento. A partir del algoritmo planteado y expresado en lenguaje Python con 192 líneas de código, se pudo desarrollar 4 tipos de topología los cuales se detalla en la siguiente tabla 8:

Tabla 8

Comparación de modelos de red LSTM

Características del modelo	Modelo 1	Modelo 2	Modelo 3	Modelo 4																																																																								
Número de unidades de procesamiento	50	100	150	200																																																																								
Épocas	200	300	400	500																																																																								
Factor de aprendizaje	0.0002	0.0003	0.0004	0.0005																																																																								
Tiempo de ejecución optimizado con GPU	35 minutos	31 minutos	33 minutos	58 minutos																																																																								
Topología de la red	<table border="1"> <tr> <td>lstm_49_input</td> <td>input:</td> <td>[(None, 21, 1)]</td> </tr> <tr> <td>InputLayer</td> <td>output:</td> <td>[(None, 21, 1)]</td> </tr> </table> <table border="1"> <tr> <td>lstm_49</td> <td>input:</td> <td>(None, 21, 1)</td> </tr> <tr> <td>LSTM</td> <td>output:</td> <td>(None, 50)</td> </tr> </table> <table border="1"> <tr> <td>dense_49</td> <td>input:</td> <td>(None, 50)</td> </tr> <tr> <td>Dense</td> <td>output:</td> <td>(None, 21)</td> </tr> </table>	lstm_49_input	input:	[(None, 21, 1)]	InputLayer	output:	[(None, 21, 1)]	lstm_49	input:	(None, 21, 1)	LSTM	output:	(None, 50)	dense_49	input:	(None, 50)	Dense	output:	(None, 21)	<table border="1"> <tr> <td>lstm_99_input</td> <td>input:</td> <td>[(None, 21, 1)]</td> </tr> <tr> <td>InputLayer</td> <td>output:</td> <td>[(None, 21, 1)]</td> </tr> </table> <table border="1"> <tr> <td>lstm_99</td> <td>input:</td> <td>(None, 21, 1)</td> </tr> <tr> <td>LSTM</td> <td>output:</td> <td>(None, 100)</td> </tr> </table> <table border="1"> <tr> <td>dense_99</td> <td>input:</td> <td>(None, 100)</td> </tr> <tr> <td>Dense</td> <td>output:</td> <td>(None, 21)</td> </tr> </table>	lstm_99_input	input:	[(None, 21, 1)]	InputLayer	output:	[(None, 21, 1)]	lstm_99	input:	(None, 21, 1)	LSTM	output:	(None, 100)	dense_99	input:	(None, 100)	Dense	output:	(None, 21)	<table border="1"> <tr> <td>lstm_149_input</td> <td>input:</td> <td>[(None, 21, 1)]</td> </tr> <tr> <td>InputLayer</td> <td>output:</td> <td>[(None, 21, 1)]</td> </tr> </table> <table border="1"> <tr> <td>lstm_149</td> <td>input:</td> <td>(None, 21, 1)</td> </tr> <tr> <td>LSTM</td> <td>output:</td> <td>(None, 150)</td> </tr> </table> <table border="1"> <tr> <td>dense_149</td> <td>input:</td> <td>(None, 150)</td> </tr> <tr> <td>Dense</td> <td>output:</td> <td>(None, 21)</td> </tr> </table>	lstm_149_input	input:	[(None, 21, 1)]	InputLayer	output:	[(None, 21, 1)]	lstm_149	input:	(None, 21, 1)	LSTM	output:	(None, 150)	dense_149	input:	(None, 150)	Dense	output:	(None, 21)	<table border="1"> <tr> <td>lstm_199_input</td> <td>input:</td> <td>[(None, 21, 1)]</td> </tr> <tr> <td>InputLayer</td> <td>output:</td> <td>[(None, 21, 1)]</td> </tr> </table> <table border="1"> <tr> <td>lstm_199</td> <td>input:</td> <td>(None, 21, 1)</td> </tr> <tr> <td>LSTM</td> <td>output:</td> <td>(None, 200)</td> </tr> </table> <table border="1"> <tr> <td>dense_199</td> <td>input:</td> <td>(None, 200)</td> </tr> <tr> <td>Dense</td> <td>output:</td> <td>(None, 21)</td> </tr> </table>	lstm_199_input	input:	[(None, 21, 1)]	InputLayer	output:	[(None, 21, 1)]	lstm_199	input:	(None, 21, 1)	LSTM	output:	(None, 200)	dense_199	input:	(None, 200)	Dense	output:	(None, 21)
lstm_49_input	input:	[(None, 21, 1)]																																																																										
InputLayer	output:	[(None, 21, 1)]																																																																										
lstm_49	input:	(None, 21, 1)																																																																										
LSTM	output:	(None, 50)																																																																										
dense_49	input:	(None, 50)																																																																										
Dense	output:	(None, 21)																																																																										
lstm_99_input	input:	[(None, 21, 1)]																																																																										
InputLayer	output:	[(None, 21, 1)]																																																																										
lstm_99	input:	(None, 21, 1)																																																																										
LSTM	output:	(None, 100)																																																																										
dense_99	input:	(None, 100)																																																																										
Dense	output:	(None, 21)																																																																										
lstm_149_input	input:	[(None, 21, 1)]																																																																										
InputLayer	output:	[(None, 21, 1)]																																																																										
lstm_149	input:	(None, 21, 1)																																																																										
LSTM	output:	(None, 150)																																																																										
dense_149	input:	(None, 150)																																																																										
Dense	output:	(None, 21)																																																																										
lstm_199_input	input:	[(None, 21, 1)]																																																																										
InputLayer	output:	[(None, 21, 1)]																																																																										
lstm_199	input:	(None, 21, 1)																																																																										
LSTM	output:	(None, 200)																																																																										
dense_199	input:	(None, 200)																																																																										
Dense	output:	(None, 21)																																																																										

De acuerdo con los dos parámetros que se han considerado para la evaluación de la Red tenemos la raíz del error cuadrático medio y el error absoluto medio. Después de haber entrenado la red se obtuvieron los siguientes resultados mostrados en la tabla 9:

Tabla 9*Comparación de las métricas de los modelos*

Métrica	Modelo 1	Modelo 2	Modelo 3	Modelo 4
RMSE_train	3.85%	3.75%	3.69%	3.57%
RMSE_val	5.81%	5.47%	5.37%	5.16%
RMSE_test	7.25%	6.77%	6.76%	6.33%
MAE_train	3.09%	2.97%	2.89%	2.75%
MAE_val	4.71%	4.33%	4.19%	3.94%
MAE_test	6.21%	5.67%	5.56%	5.07%
Accuracy_train	96.58%	96.85%	96.90%	97.08%
Accuracy_val	96.81%	96.56%	97.00%	97.14%
Accuracy_test	90.46%	91.21%	91.98%	92.35%

Elaboración: Propia

Según como se puede observar en la tabla anterior el modelo 4 obtiene la raíz del error cuadrático medio, menor al de los demás modelos en su set de testeo, en donde se probó la capacidad de generalización de la red con datos con los que no fue entrenada ni validada, así como también el error absoluto medio, y además se comparó la métrica de precisión con los demás modelos en donde se obtuvo una precisión del 92.35% por lo tanto es el modelo que se escoge para poder evaluar la tendencia de las acciones de las empresas que servirán como entrada para el algoritmo genético.

A continuación se muestra las curvas de la raíz del error cuadrático medio tanto para el set de entrenamiento y validación del modelo 4 tanto para el primer conjunto de acciones de la primera empresa cuyo nemonico es "AIHC1" (ver figura 26), así como de la última empresa cuyo nemonico es "GBV1C1" (Ver figura 27)

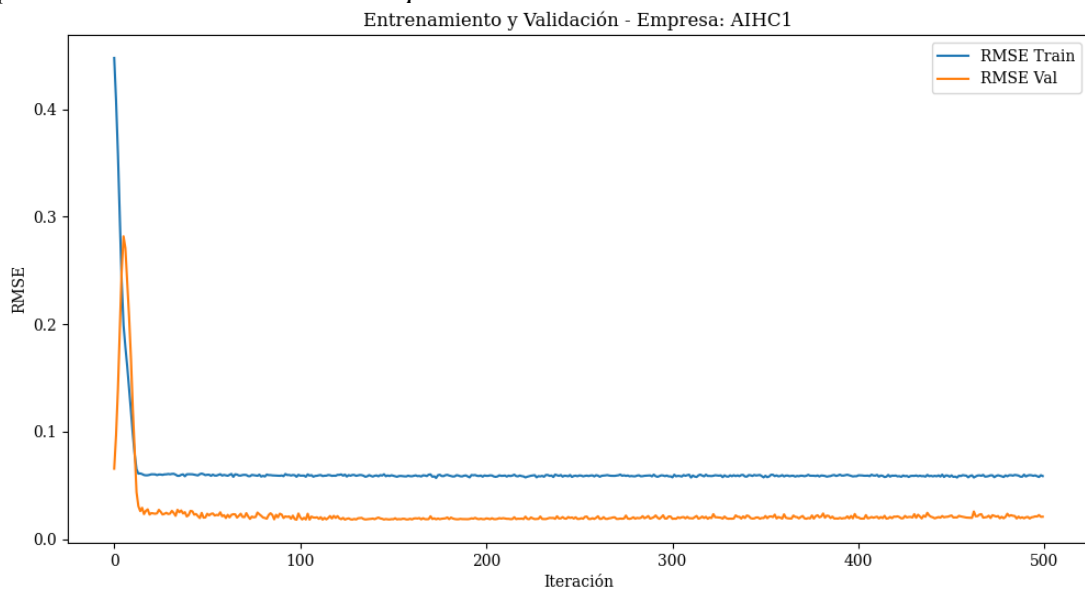
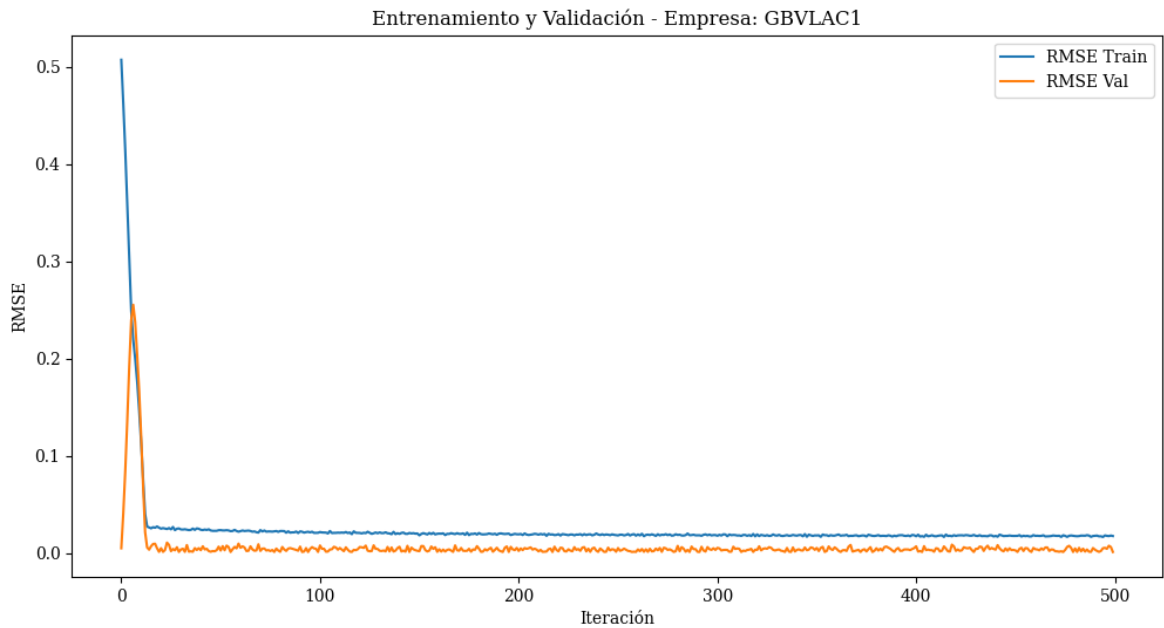
Figura 26*Representación de RMSE de la empresa AIHC1*

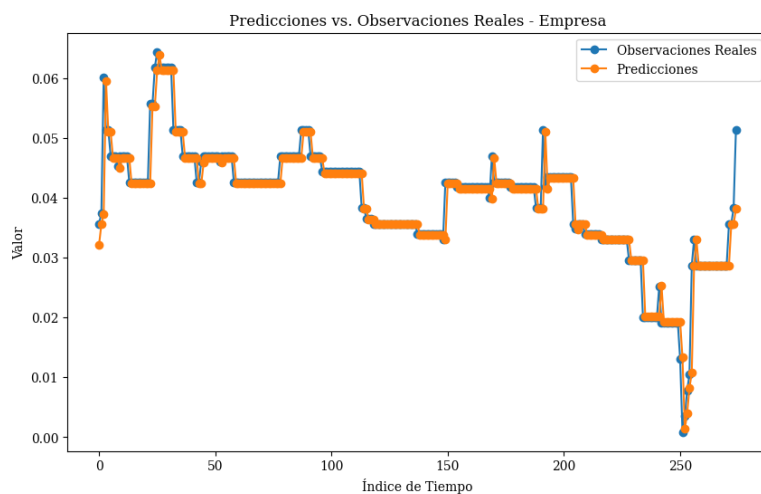
Figura 27
Representación de RMSE de la empresa GBVLAC1



Tal como se puede observar en las figuras anteriores las dos curvas siguen si tendencia decreciente sin llegar a tocarse, por lo que se puede inferir que el modelo no llega a tener overfitting, por lo que el modelo tener una generalización aceptable.

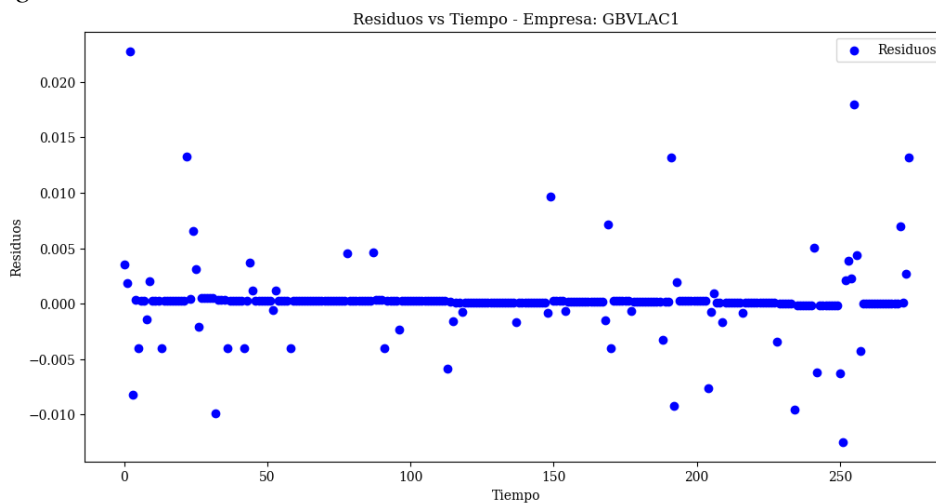
Así también mediante el uso de las librerías respectivas de visualización de datos se puede observar las líneas de tendencia en la serie de tiempo en comparativa entre las observaciones y las predicciones generadas por el modelo se ajustan de forma similar en los mismos rangos de tiempo para una empresa en específico. Esto se muestra en la figura 28

Figura 28
Comparación de datos de validación de la empresa GBVLAC1 con las predicciones del modelo



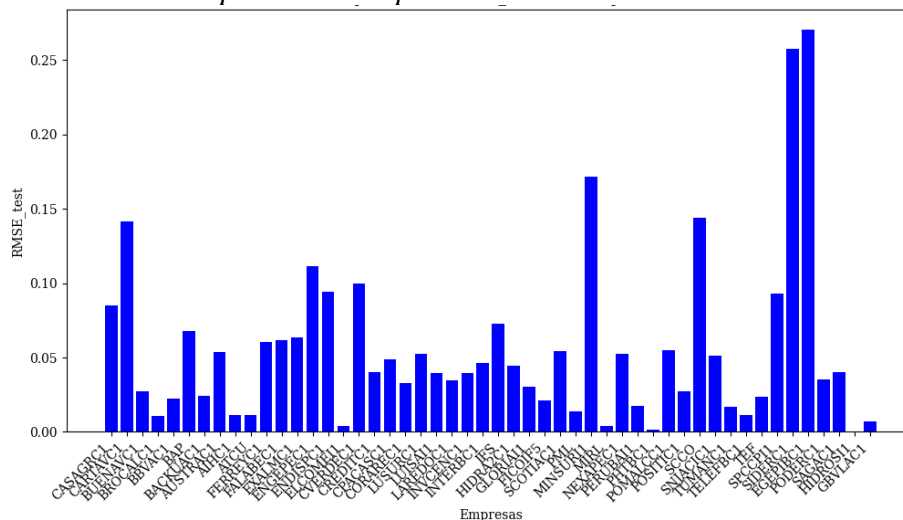
Otro elemento importante que se debe evaluar en este tipo de modelos de machine learning es el comportamiento de los residuos, y para esto se hace un análisis de la tendencia que siguen estos para el set de testeo, en donde se puede observar la tendencia preponderante acentuada en el residual medio cuyo valor es 0 con 3 decimales de precisión cuya desviación estándar respecto a la media es de 0.024 aproximadamente y de -0.014; siendo estos valores muy pequeños. (ver figura 29).

Figura 29
Residuos generados en el set de validación



Las dos graficas anteriores muestran el comportamiento para una sola empresa, sin embargo en terminos generales el comportamiento de las 50 empresas en cuanto a la raíz del error cuadratico medio se representa en un punto maximo aproximadamente de 0.03, tal como se muestra en la siguiente figura 30

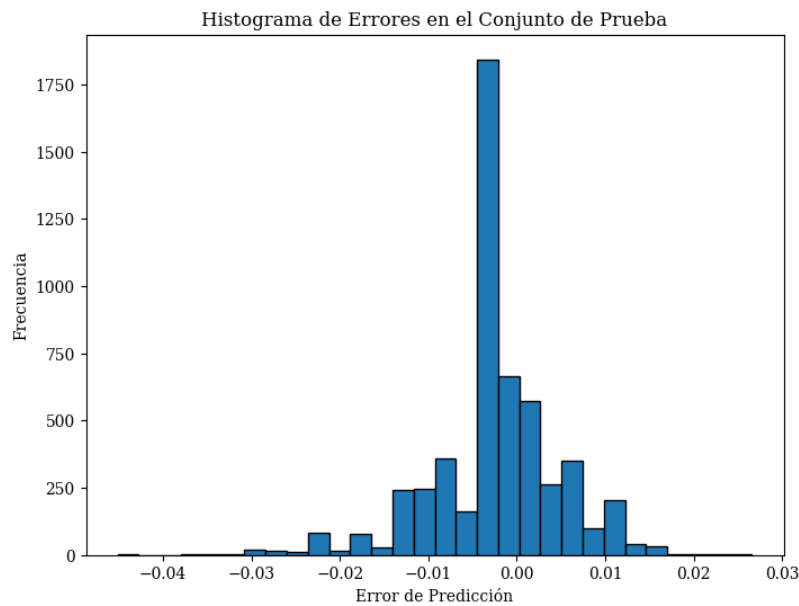
Figura 30
Representación del RMSE para cada empresa evaluada



Finalmente, como referencia grafica se analizó mediante un histograma la frecuencia respecto al error de la predicción de los modelos, en donde se puede observar en la figura 31 una clara tendencia en el valor 0 para el error de predicción.

Figura 31

Histograma de errores



Conociendo la arquitectura de red neuronal LSTM más eficiente, se procedió a usar la fórmula propuesta, $P_{(t+1)} > P_{(t)} + [\sigma_{\%Mercado}] * P_{(t)}$, en donde 24 activos de capital (Ver anexo 3) cumplieron con esta condición que garantiza que las empresas con acciones con tendencia al alza mejoren el comportamiento del portafolio de inversión, y por lo tanto son los activos que serán entradas para el algoritmo genético teniendo rendimientos próximos positivos en un periodo de 21 días.

4.1.2. Modelamiento de los elementos del algoritmo genético

La estructura del algoritmo genético debe contener una estandarización de datos para que estos puedan ser procesados, iniciando desde la estructura de los individuos o posibles soluciones mediante la representación cromosómica, así como su función de aptitud que evaluará la calidad de cada individuo para que cada uno de los candidatos con mejores características deje descendencia y se generen mejores soluciones.

Se debe considerar que para el caso de este apartado se consideró 24 activos de capital por lo que el espacio muestral se denotará bajo la siguiente representación:

$$\Omega = \{S \subseteq \{A_1, A_2, \dots, A_{24}\} / |S| \geq 1\}$$

Donde:

Ω : Espacio muestral de todos los activos de capital

S : es un subconjunto de los activos disponibles

$|S|$: Restricción que asegura que cada portafolio tenga al menos un activo

Individuos del algoritmo genético: Para la optimización del algoritmo genético cada individuo representará una solución al problema propuesto, es decir cada cromosoma contendrá los porcentajes del capital que serán distribuidos en cada uno de los activos. Para el caso de esta investigación se ha identificado 24 activos de capital que serán evaluados dentro del algoritmo genético por lo tanto los individuos serán representados por un vector con porcentajes.

Su representación es tal como se muestra a continuación:

$$C = [w_1, w_2, \dots, w_n]$$

Donde:

n : es el número de activos

w_i : el peso o representación porcentual de inversión asignado al activo i

Este vector C debe estar sujeto a las restricciones:

$$0 \leq \omega_i \leq 1 \text{ para todo } i$$

$$\sum_{i=0}^{i=n} \omega_i = 1$$

Por lo tanto, los individuos tendrán la siguiente forma cromosómica en donde cada peso asignado será un gen, esto se observa en la figura 32:

Figura 32

Representación cromosomica de los individuos

W ₁	W ₂	W ₃	W ₄	W ₅	.	.	.	W ₂₄
----------------	----------------	----------------	----------------	----------------	---	---	---	-----------------

Por lo tanto, el genotipo serán números porcentuales y el fenotipo tendrá la misma representación.

Función objetivo: Para el diseño de la función de aptitud se tomó en cuenta el avance de las diversas teorías existentes sobre portafolios de inversión, desde la teoría base de Harry Markowitz (1952) que consideró la media como rendimiento y la varianza como medida de riesgo, y que a medida que pasó el tiempo se complementó con teorías que fueron incluyendo más elementos y variables que explicaban mejor el rendimiento como el CAPM (1960) con la inclusión de la prima de riesgo, pasando por el modelo de 4 variables de Fama y French (1992), llegando a teorías más complejas como la de valoración de activos de arbitraje(1976) y en un caso particular la teoría de Black-Litterman (1992) que incluye la percepción y creencias del inversionistas frente a la incertidumbre. Todos estos enfoques tienen una tendencia de completar la teoría base, desde las variaciones sistemáticas que pueden originarse en el mercado teniendo en cuenta consideraciones muy específicas como el comportamiento del mismo, así como las primas originadas por invertir y arriesgar en un activo específico, y finalmente llegar a incluir como variables, aquellas que consideran las subjetividades del inversor.

Para el caso de la presente investigación se tomó en cuenta evaluar el portafolio optimizado a partir de 2 modelos de función objetivo con la finalidad de comparar el desempeño de cada uno, el primer modelo de función objetivo comúnmente encontrado en el estado del arte, el cual es el radio de Sharpe, que mide el rendimiento del portafolio por cada unidad de riesgo, esta función objetivo busca maximizarse durante las generaciones del algoritmo genético, y se muestra a continuación:

$$MAX(\text{Radio de Sharpe}) = \frac{R_p - R_f}{\sigma}$$

Donde:

R_p : Rendimiento del portafolio

R_f : Tasa libre de riesgo, en donde para el caso del presente estudio son representados por los bonos soberanos del Perú

σ : *Volatilidad del portafolio*

Finalmente, la segunda función objetivo se orienta a la participación de múltiples elementos o factores que representen el objetivo de la optimización ya que según las teorías existentes sugieren que hay factores sistemáticos u otros, que ayudan a generar con mejor precisión el comportamiento del rendimiento y de la volatilidad. Para el caso particular de este estudio se usaron los factores, rendimiento, volatilidad, radio de Sharpe, considerando que el inversor tendrá predisposición o confianza en un portafolio que genere rendimiento mayor por cada unidad de riesgo, prima de mercado considerando la predisposición del inversionista por invertir su capital en activos que generen mayores rendimientos de mercado, y finalmente la diversificación de portafolio como una medida que premia o penaliza a las carteras más o menos diversificadas tal como menciona Markowitz (1959) a mayor diversificación se tiende a reducir el riesgo debido a que algunos activos absorberán las volatilidades de otros.

La función objetivo se construye teniendo en cuenta los siguientes elementos:

$$MAX(\text{función Objetivo}) = \sum \lambda_i F_j$$

Donde:

λ_i : *es el peso i asignado a cada factor de la función objetivo*

F_j : *Es el factor j considerado dentro de la función objetivo*

Como se ha mencionado anteriormente el acceso a la información se limita a los datos del comportamiento de los activos, por lo que los factores rentabilidad y volatilidad estarán presentes dentro de la ecuación como factores relevantes expresadas de la siguiente manera:

$\lambda_1 R_p$: *Rendimiento del portafolio*

$-\lambda_2 \sigma$: *Volatilidad del portafolio que en este caso es negativa debido a que castiga a la función objetivo considerando que el inversor busca minimizarla*

Así también otros factores que se consideraron, es la orientación del inversor por asignar recursos a portafolios que maximicen el rendimiento por cada unidad de riesgo invertido, siendo este indicador el radio de Sharpe:

$$\lambda_3 \left(\frac{R_p - R_f}{\sigma} \right)$$

Se debe considerar para el caso de este factor una restricción específica considerando que solo se debe tener en cuenta este factor si es el que el rendimiento del portafolio es mayor al de la tasa libre de riesgo:

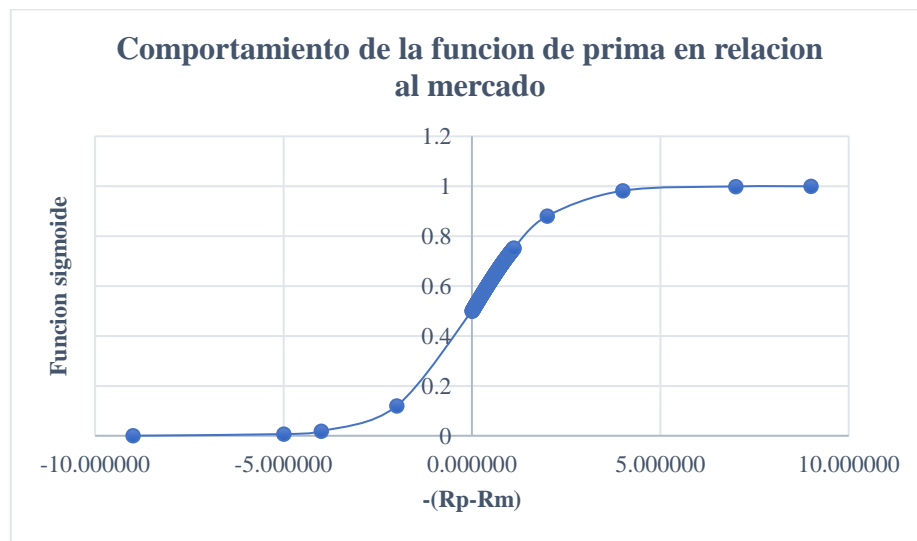
$$\text{Si } R_p \geq R_f \text{ entonces: } \lambda_3 \left(\frac{R_p - R_f}{\sigma} \right)$$

Sino: $\lambda_3 = 0$

Otro factor que se consideró relevante es la prima de riesgo sobre el mercado, en donde para este elemento al no tener una fórmula conocida se consideró usar una función sigmoide teniendo en cuenta que mientras mayor sea la diferencia entre el rendimiento del portafolio y del mercado, el valor de esta función se orientara a 1 y mientras menor se aproximara a su valor asintótico 0 tal como se muestra en la siguiente figura 33:

Figura 33

Representación de la función asociada a la prima de mercado



El comportamiento de esta función fue expresado a partir de una simulación de distribución normal según los datos en un rango para la prima de mercado entre -10 a 10, con una media de 0.55 y una desviación estándar de 0.38 siendo la prueba de bondad de ajuste de Kolmogórov-Smirnov con un p-valor de 85.57%, en donde el área sombreada resaltante muestra una concentración de valores en donde el valor del rendimiento es mayor al rendimiento de mercado, tal como se puede observar a partir de que la prima en relación al mercado se vuelve positiva los valores tienen a aumentar y aproximarse al valor de la unidad, mientras que si la diferencia es menor este se acercara a 0 por lo que

ya dentro de la función se considera la restricción de castigar a la función si es que los valores son pequeños o negativos

. Por lo tanto, la función se ajusta a lo requerido para la función objetivo y se expresaría de la siguiente manera:

$$\lambda_4 \left(\frac{1}{1 + e^{-(Rp-Rm)}} \right)$$

Finalmente el último factor a considerar es la diversificación del portafolio expresado como el promedio de la correlación de los activos que la conforman, y cuyo comportamiento está relacionado a una medida de penalización o premio a la función objetivo, ya que al ser un valor entre 0 a 1 y colocarse en el denominador, cuando la correlación es mayor tiende a castigar a la función, mientras que en valores pequeños o pocos correlacionados (portafolios más diversificados) tiende a premiar la función. La expresión matemática es la siguiente:

$$\lambda_5 \frac{2\rho_{ij}}{n(n-1)}$$

Donde

ρ_{ij} : es la correlación entre el activo i y j que conforman el portafolio

$\frac{n(n-1)}{2}$: Cantidad de combinaciones totales de activos que conforman el portafolio

conconsiderando todos los factores antes mencionados se formula la siguiente función objetivo que las contempla:

$$MAX (Función Objetivo) = \frac{\lambda_1 R_p - \lambda_2 \sigma + \lambda_3 \left(\frac{R_p - R_f}{\sigma} \right) + \lambda_4 \left(\frac{1}{1 + e^{-(Rp-Rm)}} \right)}{\lambda_5 \left(\frac{2\rho_{ij}}{n(n-1)} \right)}$$

Donde:

λ_i : Son los pesos asignados a cada factor de acuerdo a su nivel de representatividad o importancia en relación a la maximización de la función objetivo, para $i = 0,1,2,3,4,5$, el valor para los pesos asignados en un proceso de selección del algoritmo genético se estableció de acuerdo al criterio del investigador, en relación a esto se hizo una simulación de Montecarlo haciendo uso de la herramienta Risk Simulator, para verificar la sensibilidad de estos factores con una cantidad de 10000 simulaciones. Se encontró los siguientes resultados mostrados en la tabla 10:

Tabla 10

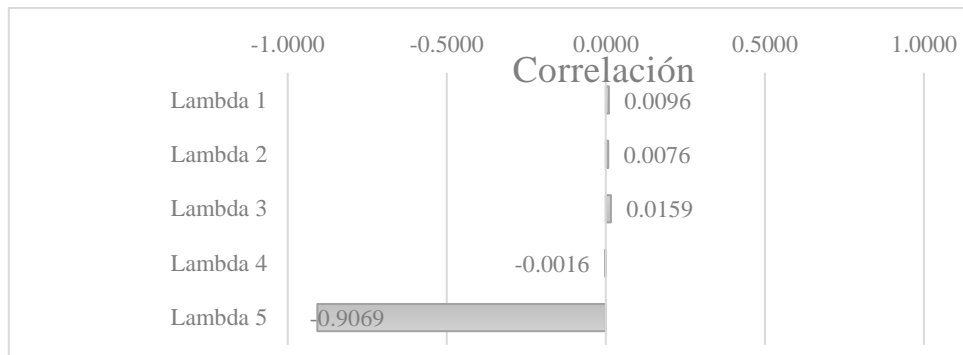
Comportamiento de las variables lambda en cuanto a su correlación y variación asociados al fitness

	Correlación	% Variación
Lambda 1	0.0096	0.01%
Lambda 2	0.0076	0.01%
Lambda 3	0.0159	0.03%
Lambda 4	-0.0016	0.00%
Lambda 5	-0.9069	82.24%

Este comportamiento se muestra en la figura 34:

Figura 34

Correlación de las variables lambda con la maximización del fitness



Para este caso en particular se usara el coeficiente encontrado como referencia para ponderar cada uno de los valores de lambda, que tal como se puede observar y en coherencia de la particularidad de cada factor asociado en donde el los lambdas asociados a la volatilidad y a la diversificación tienen valores negativos.

Se procedio a hacer la ponderación empírica de los lambdas a partir de una escala del 0 a 1 con una precisión de dos puntos flotantes, estos valores proporcionales a su nivel de correlación tal como se muestra la tabla 11:

Tabla 11

Proporción asignada de lambda's a cada factor del fitness

	Correlación	% asignado
Lambda 1	0.0096	1.0194%
Lambda 2	0.0076	0.8028%
Lambda 3	0.0159	1.6925%
Lambda 4	0.0016	0.1741%
Lambda 5	0.9069	96.3112%
Total	0.94	100.0000%

Considerando estos valores la ecuación tendrá la siguiente representación:

$$MAX(F.O) = \frac{0.0102 * R_p - 0.008 * \sigma + 0.017 * \left(\frac{R_p - R_f}{\sigma}\right) - 0.0017 * \left(\frac{1}{1 + e^{-(R_p - R_m)}}\right)}{0.963 * \left(\frac{2\rho_{ij}}{n(n-1)}\right)}$$

Hiperparámetros del algoritmo genético: La determinación de hiperparámetros es una etapa fundamental para el desarrollo, diseño y evaluación del algoritmo genético, para este caso específico se utilizó ciertas configuraciones basadas en el tipo de datos, entorno del problema y experiencia del investigador.

Tamaño de la población: Para el tamaño de la población inicial se considera evaluar el algoritmo en 3 espacios de búsqueda distintos de 1000, 2500 y 5000 individuos.

Número de generaciones: El número de generaciones estará en un rango de 50, 100 y 500; y se fueron ajustando dependiendo de la convergencia del algoritmo

Probabilidad de cruce: La probabilidad de cruce para el caso específico del algoritmo al tener números de puntos flotantes como individuos y el espacio de búsqueda muy amplio se escogió un valor de cruce de 80%.

Probabilidad de mutación: Considerando lo dicho antes en la probabilidad de cruce, se requiere que cuando el algoritmo este en un espacio de búsqueda mute con una baja probabilidad entre 0 a 10%, siendo para el caso específico 9%, el cual es el límite superior de este rango.

Operadores de selección, cruce y mutación: Para el caso específico de esta investigación se utilizó la selección por torneo, en donde se escogen una cantidad de individuos específica y de evalúan, por otra parte, el operador de cruce es el CxBlend por sus características de generar nuevos individuos que exploran ampliamente los espacios de búsqueda al generar descendientes con características de los dos padres y de forma lineal lo que se ajusta al tipo de problema planteado. Este operador tiene la siguiente representación:

$$D_i = P_{1i} + 0.8 (P_{2i} - P_{1i})$$

Donde:

D_i : Es el valor del gen i en el descendiente D

P_{1i} : Es el valor del gen i en el padre $P1$

P_{2i} : Es el valor del gen i en el padre P2

α : Es un factor de mezcla aleatorio que controla la proporción de influencia del padre P2 sobre el padre P1 en el gen i . Este factor es aleatorio y se puede ajustar dentro del rango $[0, 1]$

En el caso de la mutación se usó la mutación gaussiana con una media de 0 y una desviación estándar para este caso específico de 0.01 debido a que se requiere generar individuos con un punto porcentual de mutación, la cual escoge dentro de un rango específico de la distribución un valor que será una perturbación de mutación en cada individuo.

$$x'_i = x_i + N(0, \sigma)$$

$$x'_i = x_i + N(0, 0.01)$$

Donde:

x'_i : Es el nuevo valor del gen después de la mutación

x_i : Es el valor original del gen

$N(0, \sigma)$: Es un número aleatorio extraído de la distribución gaussiana con media 0 y desviación estándar 0.01

4.1.3. Evaluación del algoritmo genético

Para la evaluación del algoritmo genético se inició con una comparación de los valores obtenidos a partir de las dos funciones objetivos planteadas anteriormente, esto con la finalidad de escoger que algoritmo evaluar y analizar de forma determinada:

Función objetivo 1:

$$MAX(F. O) = \frac{0.0102 * R_p - 0.008 * \sigma + 0.017 * \left(\frac{R_p - R_f}{\sigma}\right) + 0.0017 * \left(\frac{1}{1 + e^{-(R_p - R_m)}}\right)}{0.963 * \left(\frac{2\rho_{ij}}{n(n-1)}\right)}$$

Función objetivo 2:

$$MAX(Radio de Sharpe) = \frac{R_p - R_f}{\sigma}$$

Después de hacer las combinaciones respectivas se muestran los resultados en la tabla 12:

Tabla 12*Comparación de funciones objetivo segun la configuración del algoritmo genetico*

Población Inicial	Número de generaciones	Función objetivo 1		Función objetivo 2	
		Rendimiento	Volatilidad	Rendimiento	Volatilidad
50	1000	0.0100054679	0.0000628696	0.0100009251	0.0000877316
100	2500	0.0100058086	0.0000612188	0.0100006323	0.0000969598
500	5000	0.0100058086	0.0000612188	0.0100009545	0.0000926398

Tal como se puede observar la función objetivo 1 muestra mejores resultados en cuanto a rentabilidad que es el valor que se ha tomado como referencia para la elección debido a que el riesgo es menor al 0.0001% para ambos casos, siendo un valor depreciable. Se debe considerar el comportamiento de los resultados mostrados en la tabla 12, donde se evidencia un aumento en el rendimiento según se aumentó la población y el número de generaciones, y de la misma forma disminuyó la volatilidad en la segunda y tercera corrida, mientras que la función objetivo dos tuvo un comportamiento estático en estos dos entornos de interacción. Por lo tanto, la función objetivo 1 planteada en esta investigación es la que se evaluó.

El modelo que tiene por función de aptitud la planteada en este estudio y se analizó mediante un diseño experimental con disposición factorial 2x3, siendo los factores el tamaño de población inicial a partir de ahora denominado TPO y el número de generación denominado NG y 3 niveles cada uno, 50, 100 y 500; y 1000, 2500 y 5000 respectivamente. Para cada uno de las combinaciones factoriales se hicieron 4 repeticiones tal como se muestra en la tabla 13:

Tabla 13*Repeticiones para cada configuración del algoritmo genetico*

NG	Tpo	Repeticiones			
		I	II	III	IV
50	1000	0.726487	0.752226	0.752226	0.752226
50	2500	0.772482	0.772482	0.772482	0.772482
50	5000	0.772482	0.772482	0.772482	0.772482
100	1000	0.752226	0.752226	0.752226	0.752226
100	2500	0.772482	0.772482	0.772482	0.772482
100	5000	0.772482	0.772482	0.772482	0.772482
500	1000	0.752226	0.752226	0.752226	0.752226
500	2500	0.772482	0.772482	0.772482	0.772482
500	5000	0.772482	0.772482	0.772482	0.772482

Después de haber generado las repeticiones se procedió a realizar en análisis de varianza con los siguientes resultados mostrados en la tabla 14:

Tabla 14
Análisis de varianza

Fuente de Variación	Gl	SC	CM	F	Sig.
NG	2	0.000037	0.000018	1	0.3811
TPo	2	0.004014	0.002007	109.0714	**0.0000
NG*TPo	4	0.000074	0.000018	1	0.4247
Error	27	0.000497	0.000018		
Total	35	0.004622			
CV	0.0024%				

Tal como se puede observar se encontró un nivel de significancia bilateral en el factor TPo, mientras que el factor NG, y la interacción entre los factores NGxTPo no tuvieron significancia estadística. También se puede evidenciar un coeficiente de variación de 0.0024% siendo un valor muy bajo lo que evidencia la estabilidad y la baja dispersión respecto a la media de los valores.

Después de encontrar los factores significativos en el diseño experimental se procedió a realizar las pruebas post-hoc, para una primera visión del comportamiento de la interacción entre factores se realizó la prueba de Tukey de diferencias de medias con un nivel de confianza del 95%. Esto se muestra en la tabla 15:

Tabla 15
Prueba de Tukey de diferencias de medias

Grupo 1	Grupo 2	Diferencias de medias	p-adj	Rechazo
100_1000	100_2500	0.0203	0	Verdadero
100_1000	100_5000	0.0203	0	Verdadero
100_1000	500_1000	0	1	Falso
100_1000	500_2500	0.0203	0	Verdadero
100_1000	500_5000	0.0203	0	Verdadero
100_1000	50_1000	-0.0064	0.4806	Falso
100_1000	50_2500	0.0203	0	Verdadero
100_1000	50_5000	0.0203	0	Verdadero
100_2500	100_5000	0	1	Falso
100_2500	500_1000	-0.0203	0	Verdadero
100_2500	500_2500	0	1	Falso
100_2500	500_5000	0	1	Falso
100_2500	50_1000	-0.0267	0	Verdadero
100_2500	50_2500	0	1	Falso

100_2500	50_5000	0	1	Falso
100_5000	500_1000	-0.0203	0	Verdadero
100_5000	500_2500	0	1	Falso
100_5000	500_5000	0	1	Falso
100_5000	50_1000	-0.0267	0	Verdadero
100_5000	50_2500	0	1	Falso
100_5000	50_5000	0	1	Falso
500_1000	500_2500	0.0203	0	Verdadero
500_1000	500_5000	0.0203	0	Verdadero
500_1000	50_1000	-0.0064	0.4806	Falso
500_1000	50_2500	0.0203	0	Verdadero
500_1000	50_5000	0.0203	0	Verdadero
500_2500	500_5000	0	1	Falso
500_2500	50_1000	-0.0267	0	Verdadero
500_2500	50_2500	0	1	Falso
500_2500	50_5000	0	1	Falso
500_5000	50_1000	-0.0267	0	Verdadero
500_5000	50_2500	0	1	Falso
500_5000	50_5000	0	1	Falso
50_1000	50_2500	0.0267	0	Verdadero
50_1000	50_5000	0.0267	0	Verdadero
50_2500	50_5000	0	1	Falso

Tal como se puede observar no existen diferencias significativas en grupos con un diferente NG, en donde el factor determinante es TPo que encuentra diferencias significativas en los niveles 1000 a 2500, mientras que para los niveles 2500 y 5000 no muestra diferencias significativas, reforzando lo encontrado en el análisis de varianza. En cuanto a la valuación de los pares de grupos dentro de este factor se realizó la prueba de Duncan para conocer mediante esta prueba más exhaustiva la diferencia de grupos dentro del nivel del factor TPo. Esto se muestra en la tabla 16:

Tabla 16
Prueba de Duncan

Contraste	A	B	dof	alternative	p-unc	BF10
Tpo	1000	2500	22	two-sided	5.45E-10	1.087e+07
Tpo	1000	5000	22	two-sided	5.45E-10	1.087e+07
Tpo	2500	5000	22	two-sided		nan

Según lo encontrado en la prueba existe diferencia significativa entre los niveles 1000 y 2500, mientras que en los niveles 2500 y 5000 no se encontró diferencia significativa. Por lo tanto, la instancia del algoritmo genético se establece con un valor NG de 50 y un TPo de 2500, esta configuración se muestra en pseudocódigo en la tabla 17:

Tabla 17

Pseudocódigo del algoritmo genético

Algorithm 1 Algoritmo Genético

- 1: **Entrada:** Librerías.
- 2: **Entrada:** Base de datos de los activos del mercado.
- 3: **Asignación de hiperparámetros:** probabilidad de cruce, probabilidad de mutación $\leftarrow 0.8, 0.09$.
- 4: **Asignación:** $R_f, R_m \leftarrow 0.002803, 0.06775$.
- 5: **Asignación:** Población inicial, Número de generaciones $\leftarrow 50, 2500$.
- 6: **Asignación:** $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5 \leftarrow 0.0096, 0.0076, 0.0159, 0.0016, 0.9069$
- 7: **Función** calcular_rendimiento_portafolio.
- 8: **Función** calcular_volatilidad.
- 9: **Función** calcular_correlación_promedio.
- 10: **Función** calcular_función_objetivo.
- 11: **Función** crear_individuo.
- 12: **Función** calcular_estabilidad.
- 13:
- 14: **Función** Algoritmo genético
- 15: Definición del problema (Maximización).
- 16: Evaluar a individuos

$$MAX(F.O) = \frac{0.0102 \cdot R_p + 0.008 \cdot \left(\frac{R_p - R_f}{\sigma}\right) + 0.017 \cdot \left(\frac{1}{1 + e^{-(R_p - R_m)}}\right) - 0.0017 \cdot \sigma}{0.963 \cdot \left(\frac{2\rho_{ij}}{n(n-1)}\right)} \quad (1)$$

17: Realizar Cruce CxBlend.

$$D_t = P_{1t} + 0.8 \cdot (P_{2t} - P_{1t}) \quad (2)$$

18: Realizar mutación Gaussiana.

$$mutGaussian(x, \mu, \sigma, pb) = x + N(0, .01^2) \cdot 0.09 \quad (3)$$

19: Realizar selección por torneo.

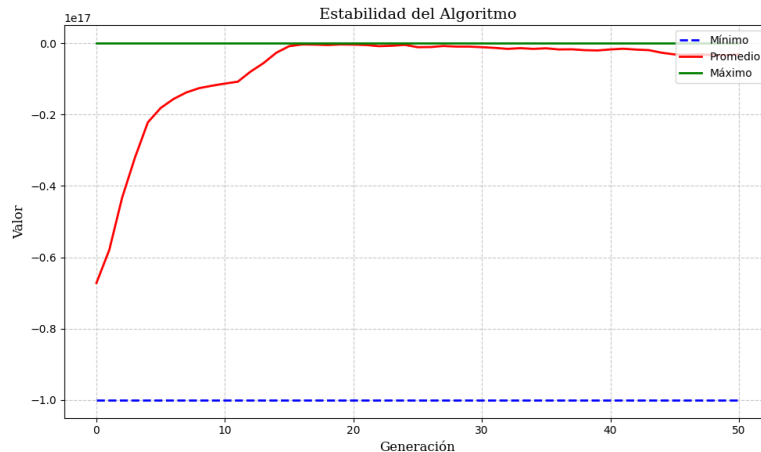
$$selTournament_3(population, tournsize) = Mejor_1 \cup Mejor_2 \cup Mejor_3 \quad (4)$$

20: **Imprimir** el mejor fitness y el mejor individuo.

Así mismo se obtuvo como resultado un fitness de 0.772482 cuya estabilidad de muestra en la siguiente figura 35:

Figura 35

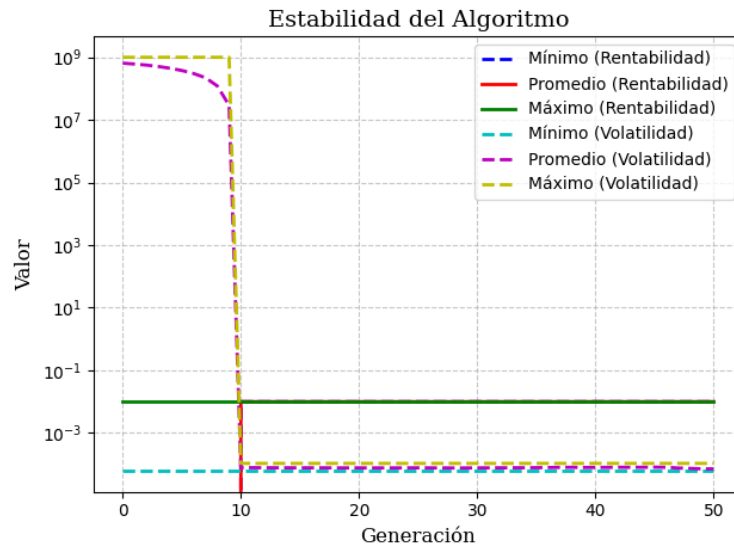
Estabilidad del algoritmo en relacion a su fitness



Se puede evidenciar que al ser un problema de maximización del fitness este se mantiene constante desde la primera interacción, así también los valores promedios se mantienen cercanos al máximo. Teniendo en cuenta que este es el valor del fitness y que este tiene asociados el rendimiento cuyo valor asociado es de 1.0005808590288486% y volatilidad de 0.00612188%. Se muestra en la siguiente figura 36, el comportamiento de estos dos valores a lo largo de las generaciones:

Figura 36

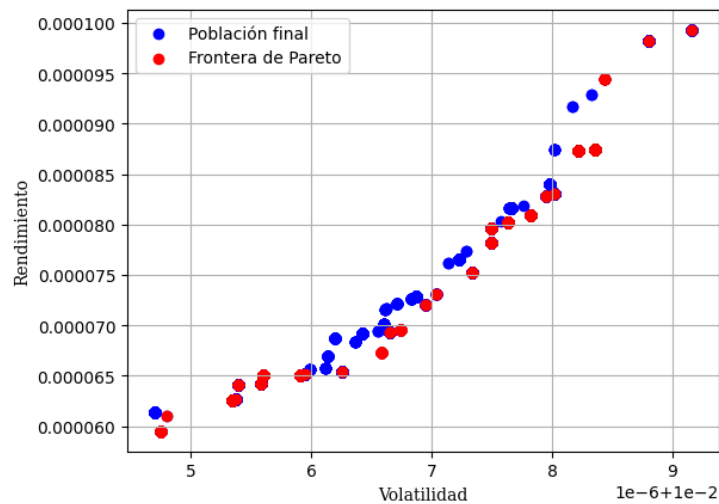
Estabilidad del algoritmo en relacion a su rentabilidad y volatilidad



Se puede observar en la figura que tanto la rentabilidad como la volatilidad encuentran convergencia desde la décima generación, en el caso la rentabilidad su estabilización es más acentuada, mientras que para el caso de la volatilidad el algoritmo debe cambiar el espacio de búsqueda a varias décimas para poder lograr la convergencia. Teniendo en cuenta que estas variables son las que se contraponen al ser la rentabilidad el elemento a maximizar y la volatilidad el elemento a minimizar, se generó la frontera de Pareto para visualizar el comportamiento de estas variables a través de las generaciones. Esto demuestra en la figura 37:

Figura 37

Frontera de pareto rendimiento vs volatilidad



Tal como menciona Markowitz (1959), estos dos elementos se contraponen, y asumir una mayor rentabilidad se debe asumir un mayor riesgo o volatilidad por lo que la frontera de Pareto antes mostrada evidencia este comportamiento que se ajusta a la naturaleza de estas variables, y no se muestra ninguna anomalía en su relación lineal

Finalmente, al conocer la estabilidad del algoritmo se puede representar al individuo que genera la optimalidad en este punto es el representado en la siguiente tabla 18:

Tabla 18

Resultados de porcentajes a invertir en cada activo perteneciente al portafolio

% a invertir	Empresas
0.2948%	CASAGRC1
6.1901%	BUENAVC1
2.5055%	BBVAC1
2.2108%	ATCU
7.0744%	FALABEC1
4.1268%	ELCOMEI1
1.0317%	CVERDEC1
4.6426%	CREDITC1
0.1474%	CPACASC1
6.4849%	CORAREC1
4.2741%	LUISAI1
5.0847%	INVCENC1
6.9270%	INTERBC1
2.8740%	IFS
4.4952%	SCOTIAC1
2.2845%	PML
1.7686%	MINSURI1
3.2424%	MIRL
2.5792%	NEXAPEC1
5.1584%	PERUBAI1
6.6323%	POSITIC1
6.1164%	TELEFBC1
7.0007%	TEF
6.8534%	SAGAC1

Tal como se puede observar las empresas con el mayor porcentaje a invertir fueron las representadas por FALABEC1 y TEF, con un 7% aproximadamente, mientras que las empresas que contempla el menor porcentaje a invertir fue CPACASC1 con un 0.1474%

4.2. DISCUSIÓN

Para la discusión de la presente investigación con diversos estudios, se debe considerar que cada autor dentro de los antecedentes y la presente investigación ha desarrollado su estudio en un entorno particular, es decir los resultados obtenidos referente a la rentabilidad o volatilidad depende específicamente de los activos de capital escogidos y evaluados por los autores.

Maholi et al. (2019) de forma similar a esta investigación usa una metodología en donde primero se usa una red neuronal artificial. En el caso del autor la arquitectura usada es de una red densa que sirve en la fase de selección de activos de capital más aptos para conformar los portafolios generados por el algoritmo genético, el cual tendrá como función de aptitud el retorno por sobre el riesgo, el cual denomina ERB, que un símil del radio de Sharpe y que mide el exceso de rentabilidad frente a una tasa base. La principal crítica respecto a esta metodología es que las redes neuronales densas no discernen en cuanto a las series de tiempo, es decir las entradas no son diferenciadas por lo que los datos se adaptan más a la estructura interna que tienen, en vez de dar una respuesta en relación a la variable tiempo en donde la precisión alcanza un 98% para un modelo unistep, el cual es inferior a esta investigación en donde este modelo tiene una precisión del 99% en una red LSTM, mientras que el MAE obtenido por el investigador es de 5.60% mientras que para la presente investigación fue de 5.07% en modelo multistep. Entre los resultados respecto al retorno del portafolio, este alcanza un 1.42% sin embargo la volatilidad del portafolio es mucho mayor en puntos porcentuales alcanzando un 1.86% siendo muy superior al 0.00612188% encontrado en el portafolio óptimo de esta investigación.

Rodríguez et al. (2020) en su investigación utiliza otro tipo de activos los cuales son las monedas de diversos países, para minimizar el riesgo debido a la estabilidad que estas representan, estas fueron codificadas en cuanto a su rentabilidad usando logaritmos neperianos. Para el caso se los operadores genéticos utilizan en forma análoga el cxblend y la mutación gaussiana, sin embargo, la función objetivo está relacionada a la convencional y usada en muchas fuentes, la cual es la asociada a la rentabilidad por cada grado de riesgo asumido. Para el caso de esta investigación difiere en el resultado de la ponderación. Para el caso de los resultados cuantificados, se obtuvo un 0.049292% de rendimiento, valor mucho menor a la propuesta de esta investigación mientras que el riesgo o volatilidad fue de 0.0354% mucho mayor que la volatilidad encontrada en la presente investigación.

Bo Liu (2023) en su investigación genera un enfoque de red neuronal compleja para optimizar el portafolio de inversión, en donde la entropía de la red se usa como medida para conocer el grado de relevancia de cada acción que conforma el portafolio que pasara a ser optimizado por un algoritmo genético, esto es un análogo del preprocesamiento de esta investigación, en donde se usa una red neuronal para seleccionar las acciones con mejor desempeño en sus próximos valores para pasar a la optimización del algoritmo genético, tal como se mencionó la principal métrica de la investigación es la entropía de la red que representa el portafolio de inversión, mientras que en la presente investigación se usó un método más relacionado al comportamiento sistemático del mercado. Los resultados encontrados con esta presente investigación difieren en cuanto a la rentabilidad siendo el resultado mayor, ya que para Bo Liu (2023) solo encontró un rendimiento de 0.2667% de retorno. Por otra parte, Chun-Hao et al. (2019) en su investigación en contraposición a esta investigación es la consideración de múltiples aspectos en la función objetivo en donde se tiene en cuenta que el portafolio este balanceado y la satisfacción relacionado al portafolio, así como métricas de inversión teniendo finalmente una fitness de 4 elementos en su numerador y un elemento en su denominador, para el caso de esta investigación otro aspecto que difiere es el número de elementos que conforman la cartera, en donde solo se evaluó 5 activos de capital, basándose principalmente en el rendimiento sobre la inversión obteniendo como resultados un máximo de 0.6% de retorno siendo inferior al rendimiento obtenido a los resultados encontrados en esta investigación.

Vasiani (2020) en su investigación considera un índice de prioridad por cada sector empresarial a evaluar por lo que se considera un preprocesamiento que lleva a los activos con mayor puntuación por cada sector para luego pasar a la optimización del portafolio de inversión, este proceso difiere al de la presente investigación teniendo en cuenta que no toma series de tiempo sino los valores de comportamiento interno de la empresa en cuanto a sus transacciones, como la compra, venta de activos o devaluación de los mismo, otro aspecto resaltante es la función objetivo en donde se usó el retorno por cada unidad de riesgo, para este caso tuvo un promedio de 3% superior al de esta investigación, sin embargo no se hace referencia al portafolio de inversión en relación a la volatilidad de los activos por lo que esta rentabilidad podría estar relacionada a un riesgo muy alto de inversión.

CAPITULO V: CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- Se desarrolló una red neuronal artificial para predecir tendencias en activos financieros, configurada con 150 unidades de procesamiento neuronales LSTM, entrenada durante 400 épocas y ajustada con una tasa de aprendizaje de 0.0004. Los resultados demostraron una precisión del 92.35%, posicionándola como una herramienta confiable de predicción, considerando que la cantidad de pasos diarios a predecir fue de 21 días. Además, se evaluaron métricas como la raíz del error cuadrático medio con un valor de 6.33% y un error absoluto medio con un valor de 5.07%, lo que indica una buena capacidad de generalización del modelo para generar predicciones correctas frente al registro de 138,200 datos analizados. Esta alta precisión y los bajos niveles de error del modelo se traducen en una mayor capacidad para identificar oportunidades de inversión rentables y minimizar la exposición a la volatilidad del mercado, asignando de forma óptima porcentajes de inversión mediante el vector resultante del algoritmo genético.
- Se modeló la función objetivo que tiene como elementos la rentabilidad, la volatilidad, el ratio de Sharpe, la prima de mercado y la diversificación del portafolio lo cual implica que el entorno en el que los individuos, que fueron posibles soluciones, se optimizaron en un contexto que contempla varios elementos de la teoría de portafolios, así también la representación cromosómica de individuos fue de un vector de valores porcentuales de capital que fueron asignados a cada activo que diversificará la cartera de inversión; y finalmente, los hiperparámetros con tres niveles de población inicial y número de generaciones, así como la probabilidad de cruzamiento de 0.8 y de mutación de 0.09, lo cual asegura la exploración y explotación del universo muestral de portafolios óptimos.
- Se encontró como mejor configuración del algoritmo genético la arquitectura que contempla 50 generaciones, lo cual asegura un bajo costo computacional del algoritmo. Así también se encontró el portafolio óptimo con 2500 individuos de población inicial asegurando una gran cantidad de exploración del universo muestral, al haberse encontrado óptimo el portafolio con una cantidad menor a 5000 individuos, además se encontró un fitness de 0.772482, una rentabilidad de 1.00058% que se traduce en un 12.00696 % anual, lo cual evidencia la estabilidad del valor de los activos de capital escogidos mediante la red neuronal al tener una precisión del 92%, teniendo en cuenta que estos activos absorben la volatilidad del mercado, por lo que se asegura la estabilidad total del portafolio lo cual se refuerza con una volatilidad muy baja de 0.00612188%.

5.2. RECOMENDACIONES

- El algoritmo genético es un modelo generalizado de optimización por lo que otros mercados de acciones de valores extranjeros podrían demostrar un atractivo mayor en términos de rentabilidad y volatilidad al ser procesados por este mismo modelo.
- Se debe considerar optimizar los tiempos de procesamiento al usar herramientas tecnológicas más potentes y minimizar el costo computacional al procesar carteras de inversión.
- Si existiera fuentes de información o bases de datos internas de las empresas se debe considerar usar las variables asociadas a la rentabilidad con la finalidad de mejorar la precisión tanto en la red neuronal como el algoritmo genético.
- Se debe incentivar a las universidades públicas y privadas a implementar en sus investigaciones el uso de herramientas de inteligencia artificial con la finalidad de obtener en términos de recursos tecnológicos, resultados favorables y que ayuden a los estudiantes a generar mejoras eficientes en distintos campos de la ciencia.

BIBLIOGRAFIA

- Álvarez García, R. D., Ortega Oliveros, G., Sánchez Ospina, A. M., & Herrera Madrid, M. (2004). *Evolución de la Teoría económica de las finanzas: Una breve revisión*. 23. <https://econpapers.repec.org/article/col000217/005593.htm>
- Ames Santillán, J. C. (2012). *Alternativas de diversificación internacional para portafolios de acciones de la Bolsa de Valores de Lima*. 7, 13. <https://www.redalyc.org/articulo.oa?id=281623577003>
- Arias Gonzáles, J. L. (2021). *Diseño y metodología de la investigación* (1st ed.).
- Baca Urbina, G. (2016). *Evaluación de proyectos* (Mc Graw Hill, Ed.).
- Basogain Olabe, X. (1998). *Redes Neuronales Artificiales y sus Aplicaciones* (Issue 2). https://ocw.ehu.eus/pluginfile.php/40137/mod_resource/content/1/redes_neuro/contenidos/pdf/libro-del-curso.pdf
- Bennett, D. (2001). How can I deal with missing data in my study? *Australian and New Zealand Journal of Public Health*, 25(5), 464–469. <https://doi.org/https://doi.org/10.1111/j.1467-842X.2001.tb00294.x>
- Bo, L. (2023). Portfolio Optimization Based on Complex Networks and Genetic Algorithms. *Academic Journal of Computing & Information Science*, 6(10). <https://doi.org/10.25236/ajcis.2023.061013>
- Cáceres Chian, V. A. E. (2018). *Predicción de precios de acciones de bolsa de valores utilizando support vector regression*.
- Candía Garcia, C. D., López Castro, L. F., & Jaimes Suárez, S. A. (2020). Selección óptima del portafolio de proyectos utilizando metaheurísticas de población y trayectoria meta-optimizadas. *Revista EIA*, 17(34), 1–18. <https://doi.org/10.24050/reia.v17i34.1399>
- Carmona Muñoz, D. M., & Vera Leyton, M. (2017). *Evaluación de los factores de riesgo en los activos de renta variable que conforman el índice S&P MILA 40: aplicación del modelo de tres factores de Fama y French en el periodo 2009-2013*. 18. <https://www.redalyc.org/pdf/3235/323553607005.pdf>
- Chen, C. H., Lu, C. Y., Hong, T. P., Lin, J. C. W., & Gaeta, M. (2019). An Effective Approach for the Diverse Group Stock Portfolio Optimization Using Grouping Genetic Algorithm. *IEEE Access*, 7, 155871–155884. <https://doi.org/10.1109/ACCESS.2019.2949055>
- Cheong, D., Kim, Y. M., Byun, H. W., Oh, K. J., & Kim, T. Y. (2017). Using genetic algorithm to support clustering-based portfolio optimization by investor information. *Applied Soft Computing*, 61, 593–602. <https://doi.org/10.1016/j.asoc.2017.08.042>
- Chkalova, O., Efremova, M., Kiryushin, S., & Muranova, T. (2020). Management system for innovation and investment projects in trade. *IOP Conference Series: Materials Science and Engineering*, 940(1). <https://doi.org/10.1088/1757-899X/940/1/012060>
- Conti, D., Simó, C., & Rodríguez, A. (2005). Teoría de carteras de inversión para la diversificación del riesgo: enfoque clásico y uso de redes neuronales artificiales (RNA)

- Investment portfolio theory and risk diversification: classic and neural networks methodology. *Revista Ciencia e Ingeniería*, 26(1), 35–42. <http://www.redalyc.org/articulo.oa?id=507550773006>
- Dagnino, J. (2014). Datos faltantes (Missing values). *Revista Chilena de Anestesia*, 43, 332–334. <https://revistachilenadeanestesia.cl/datos-faltantes-missing-values/>
- Darwin, C. (1859). *On the origin of species by means of natural selection or the preservation of favoured races in the Struggle for life* (Revised Edition).
- Das, A., Chaudhuri, T., Roy, S. S., Biswas, S., & Guha, B. (2023). Selection of Appropriate Portfolio Optimization Strategy. *Theoretical and Applied Computational Intelligence*, 1(1), 58–81. <https://doi.org/10.31181/taci1120237>
- de Greiff, S., & Rivera, J. C. (2018). Investment portfolio optimization with transaction costs through a multiobjective genetic algorithm: An applied case to the Colombian Stock Exchange. *Estudios Gerenciales*, 34(146), 74–87. <https://doi.org/10.18046/j.estger.2018.146.2812>
- De Sousa Santana, F. (2013). *Modelo de valoración de activos financieros (CAPM) y teoría de valoración por arbitraje (APT): Un test empírico en las empresas del sector eléctrico brasileño**. 16.
- Didier, T., Levine, R., Llovet Montanes, R., & Schmukler, S. L. (2021). Capital market financing and firm growth. *Journal of International Money and Finance*, 118. <https://doi.org/10.1016/j.jimonfin.2021.102459>
- Fang, C., & Wang, C. (2020). *Time Series Data Imputation: A Survey on Deep Learning Approaches*. <http://arxiv.org/abs/2011.11347>
- García Serrano, Alberto. (2017). *Inteligencia artificial: fundamentos, práctica y aplicaciones*. Alfaomega.
- Gitman, L. J., & Joehnk, M. D. (2009). *Fundamentos de inversiones*. Pearson Educación.
- Goldberg, D. (1989). *Genetic Algorithms in search, optimization & Machine Learning*. Addison-Wesley Publishing Company, INC.
- Gorgolis, N., Hatzilygeroudis, I., Istenes, Z., & Grad Gyenne, L. (2019). Hyperparameter Optimization of LSTM Network Models through Genetic Algorithm. *International Conference on Information, Intelligence, Systems and Applications*. <https://doi.org/10.1109/IISA.2019.8900675>
- Granada Echeverri, M., & Santa Chávez, J. J. (2016). *Optimización combinatoria-De la teoría a la practica* (1st ed.).
- Gutiérrez Reina, D., Tapia Córdoba, A., & del Nozal, Á. (2020). *Algoritmos Genéticos con Python* (Primera). Alfaomega-Marcombo.
- Hart, O. D., & Zingales, L. (2011). Inefficient provision of liquidity. *NBER Working Paper Series*. <http://www.nber.org/papers/w17299>
- Hernández Sampieri, R., & Mendoza Torres, C. P. (2018). *Metodología de la investigación: Las rutas cuantitativa, cualitativa y mixta* (1st ed.).

- Hewamalage, H., Bergmeir, C., & Bandara, K. (2021). Recurrent Neural Networks for Time Series Forecasting: Current status and future directions. *International Journal of Forecasting*, 37(1), 388–427. <https://doi.org/10.1016/j.ijforecast.2020.06.008>
- Isasi Viñuela, P., & Galván León, I. (2004). *Redes neuronales artificiales* (Pearson Educación).
- Jiménez Carrión, M. (2016). *Los algoritmos genéticos desde la investigación de operaciones* (1st ed., Vol. 1). Editorial Académica Española.
- Li, C., Shen, L., & Qian, G. (2023). Online Hybrid Neural Network for Stock Price Prediction: A Case Study of High-Frequency Stock Trading in the Chinese Market. *Econometrics*, 11(2). <https://doi.org/10.3390/econometrics11020013>
- Li, Q., Al-Jaifi, H. A., & Kamaruddin, N. (2023). *Can Long-short Term Memory Neural Network With Symbolic Genetic Algorithm Predict Stock Price Change Basing on Fundamental Indicators*. <https://doi.org/10.21203/rs.3.rs-3306087/v1>
- Liagkouras, K. (2019). A new three-dimensional encoding multiobjective evolutionary algorithm with application to the portfolio optimization problem. *Knowledge-Based Systems*, 163, 186–203. <https://doi.org/10.1016/j.knosys.2018.08.025>
- Lim, S., Kim, M. J., & Ahn, C. W. (2020). A Genetic Algorithm (GA) Approach to the Portfolio Design Based on Market Movements and Asset Valuations. *IEEE Access*, 8, 140234–140249. <https://doi.org/10.1109/ACCESS.2020.3013097>
- López Rojas, J. F., Viasus Carrasquilla, O., Pérez Castañeda, D., Silva Bello, C. D. M., & Martínez Patiño, M. A. (2015). *Optimización de portafolios mediante el modelo de Black-Litterman supervisado por la Teoría de Control*. 14. <http://polux.unipiloto.edu.co:8080/00002509.pdf>
- Maholi Solin, M., Alamsyah, A., Rikumahu, B., & Arya Saputra, M. A. (2019). Forecasting Portfolio Optimization using Artificial Neural Network and Genetic Algorithm. *IEEE Xplore*, 1–7. <https://doi.org/10.1109/ICoICT.2019.8835344>
- Maldonado, C. E. (2013). *Un problema fundamental en la investigación: Los problemas P vs. NP*. 12. <https://www.redalyc.org/pdf/5177/517751544002.pdf>
- Markowitz, H. (1952). Portfolio Selection. *Journal of Economic Literature*, 7(1), 77–91. <http://links.jstor.org/sici?sici=0022-1082%28195203%297%3A1%3C77%3APS%3E2.0.CO%3B2-1>
- McCall, J. (2005). Genetic algorithms for modelling and optimisation. *Journal of Computational and Applied Mathematics*, 184(1), 205–222. <https://doi.org/10.1016/j.cam.2004.07.034>
- Mousapour Mamoudan, M., Ostadi, A., Pourkhodabakhsh, N., Fathollahi-Fard, A. M., & Soleimani, F. (2023). Hybrid neural network-based metaheuristics for prediction of financial markets: A case study on global gold market. *Journal of Computational Design and Engineering*, 10(3), 1110–1125. <https://doi.org/10.1093/jcde/qwad039>
- Patterson, J., & Gibson, A. (2017). *Deep Learning* (M. Loukides & T. McGovern, Eds.).

- Pilbeam, K. (2005). The Capital Asset Pricing Model. In *Finance and Financial Markets* (Segunda, p. 479). Palgrave Macmillan.
- Ponce Cruz, P. (2010). *Inteligencia artificial con aplicaciones a la ingeniería* (A. Herrera, Ed.; Primera Edición).
- Pratama, I., Erna Permanasari, A., Ardiyanto, I., & Indrayani, R. (2016). *A Review of Missing Values Handling Methods on Time-Series Data*.
- Reyes Osorio, T. I. (2023). *Evaluacion-de-metodologias-de-imputacion-de-datos-en-motores-diesel-para-el-desarrollo-de-sistemas-de-diagnostico-inteligente-de-fallas*. <https://repositorio.uchile.cl/bitstream/handle/2250/193770/Evaluacion-de-metodologias-de-imputacion-de-datos-en-motores-diesel-para-el-desarrollo-de-sistemas-de-diagnostico-inteligente-de-fallas.pdf?sequence=1>
- Rodríguez López, E., García Mejía, J. F., & Martínez Garduño, Y. (2020). Diseño de un portafolio de inversión con algoritmos evolutivos. *Saeta Digital*, 6(1), 51–70.
- Ruo Chen, L., & Muchao, L. (2021). Stock Trend Prediction Algorithm Based on Deep Recurrent Neural Network. *Wireless Communications and Mobile Computing*, 2021. <https://doi.org/10.1155/2021/5694975>
- Serrano, A., Soria, E., & Martín, J. (2010). *Redes Neuronales Artificiales*. http://ocw.uv.es/ingenieria-y-arquitectura/1-2/libro_ocw_libro_de_redes.pdf
- Siami-Namini, S., Tavakoli, N., & Siami Namin, A. (2019). *The Performance of LSTM and BiLSTM in Forecasting Time Series USA*. [https://doi.org/978-1-7281-0858-2/19/\\$31.00](https://doi.org/978-1-7281-0858-2/19/$31.00)
- Vasiani, V. D., Handari, B. D., & Hertono, G. F. (2020). Stock portfolio optimization using priority index and genetic algorithm. *Journal of Physics: Conference Series*, 1442(1). <https://doi.org/10.1088/1742-6596/1442/1/012031>
- Velasco Reyes, J. A., & Mejía Sanabria, E. D. (2019). *Modelo APT (arbitrage pricing theory) para el análisis de la rentabilidad esperada a partir de los factores de riesgos macroeconómicos a los que se expone el mercado de capitales colombiano*. https://repository.unab.edu.co/bitstream/handle/20.500.12749/13646/2019_Tesis_Velasco_Reyes_Juan_Alberto.pdf?sequence=1&isAllowed=y
- Vélez, M. C., & Montoya, J. A. (2007). *METAHEURÍSTICOS: UNA ALTERNATIVA PARA LA SOLUCIÓN DE PROBLEMAS COMBINATORIOS EN ADMINISTRACIÓN DE OPERACIONES*. 18. <https://www.redalyc.org/pdf/1492/149216950008.pdf>
- Vorobioff, J., Cerrotta, S., Amadio, A., & Morel, E. (2022). *Inteligencia Artificial y Redes Neuronales Fundamentos, Ejercicios y Aplicaciones* (Primera Edición). <https://www.researchgate.net/publication/359716455>

ANEXOS

Anexo 1 Matriz de consistencia

TÍTULO: “ALGORITMO GENÉTICO Y RED NEURONAL ARTIFICIAL PARA LA OPTIMIZACIÓN DE PORTAFOLIOS DE INVERSIÓN”						
PROBLEMA	OBJETIVOS	HIPÓTESIS	VARIABLES	INDICADORES	METODOLOGÍA DE LA INVESTIGACIÓN	
¿Es posible optimizar un portafolio de inversión usando algoritmos genéticos y una red neuronal artificial?	Diseñar un algoritmo genético y red neuronal artificial para optimizar los portafolios de inversión			Error Absoluto medio	Ámbito de estudio Al realizar la evaluación de activos de capital de la Bolsa de Valores de Lima, el alcance del estudio a nivel geográfico abarca el Perú, considerándose que los datos de entrada que son valores cuyo entorno en el que se definen está dentro del ámbito nacional.	
¿Cuál es la arquitectura de la red neuronal artificial para predecir la tendencia de los activos de capital que se usarán en el algoritmo genético?	Diseñar una red neuronal artificial para predecir la tendencia de los activos de capital que usará el algoritmo genético	El algoritmo genético y la red neuronal artificial diseñados optimizan el portafolio de inversión	Red Neuronal artificial	Raíz del error cuadrático medio	Método de investigación El método de investigación es de tipo hipotético deductivo Unidad de análisis La unidad de análisis son los activos de capital en el mercado de valores nacional, que cotizan en la Bolsa de Valores de Lima.	
¿Cuál es el modelamiento de la función objetivo, individuos e hiperparámetros del algoritmo genético para la optimización del portafolio de inversión?	Modelar la función objetivo cromosómica de individuos e hiperparámetros a evaluar en el algoritmo			Rendimiento del portafolio	Población y Muestra Población: 255 activos de capital de la Bolsa de Valores de Lima	
¿Cuál configuración del algoritmo genético es más eficiente para optimizar el portafolio de inversión?	Evaluar las diversas configuraciones del algoritmo genético para determinar su eficiencia			Riesgo del portafolio	Muestra: 255 activos de capital de la Bolsa de Valores de Lima	

Anexo 2

Listado de empresas evaluadas con nemónicos

Nemónico	Nombre de la Empresa	Nemónico	Nombre de la Empresa
CASAGRC1	Casa Grande S.A.A.	INTERBC1	Interbank
CARTAVC1	Cartavio S.A.A.	IFS	Intercorp Financial Services Inc.
BUENAVC1	Compañía de Minas Buenaventura S.A.A.	HIDRA2C1	Hidrandina S.A.
BROCALC1	Sociedad Minera El Brocal S.A.A.	GLORIAI1	Gloria S.A.
BBVAC1	BBVA Continental	FICOIF5	Financiera Confianza S.A.A.
BAP	Credicorp Ltd.	SCOTIAC1	Scotiabank Perú S.A.A.
BACKUAC1	Banco de Crédito del Perú	PML	Panoro Minerals Ltd.
AUSTRAC1	Austral Group S.A.A.	MINSURI1	Minera IRL Ltd.
AIHC1	Inversiones Centenario S.A.A.	MIRL	Minera IRL Ltd.
ATCU	Andino Investment Holding	NEXAPEC1	Nexa Resources Perú S.A.A.
FERREYC1	Ferreycorp S.A.A.	PERUBAI1	Empresa Agraria Azucarera Andahuasi S.A.
FALABEC1	Falabella Perú S.A.	PHTBC1	Perú Holding de Turismo S.A.
EXALMC1	Exalmar S.A.A.	POMALCC1	Minera Pomalca S.A.A.
ENGEPEC1	Engie Energía Perú S.A.	POSITIC1	Positiva Seguros y Reaseguros
ENDISPC1	Enel Distribución Perú S.A.A.	SCCO	Southern Copper Corporation
ELCOMEI1	El Comercio S.A.	SNJACIC1	San Jacinto S.A.A.
CVERDEC1	Cementos Pacasmayo S.A.A.	TUMANC1	Empresa Agroindustrial Tumán S.A.
CREDITC1	Grupo Crédito S.A.	TELEFBC1	Telefónica del Perú S.A.A.
CPACASC1	Cementos Pacasmayo S.A.A.	TEF	Telefónica S.A.
CORAREC1	Corporación Aceros Arequipa S.A.	SPCCPI1	Southern Peru Copper Corporation
LUSURC1	Luz del Sur S.A.A.	SIDERC1	Siderperú S.A.A.
LUISAI1	Luis A. Heller S.A.C.	EGEPIBC1	Egepi Bocamina S.A.
LAREDOC1	Larcomar Edificaciones S.A.	PODERC1	Empresa Eléctrica de Piura S.A.
INVCENC1	Inversiones Centenario S.A.A.	SAGAC1	Sociedad Agrícola Ganadera El Álamo S.A.
HIDROSI1	Hidrandina S.A.	GBVLAC1	Bolsa de Valores de Lima

Anexo 3

Datos de rendimientos de empresas seleccionadas

data_time	CASAGRC1	CARTAVCI	BUENAVCI	BROCALCI	BBVAC1	BAP	BACKUAC1	AUSTRAC1	LUISAI1	LAREDOC1	PHTBC1	POMALCC1	POSITIC1	SCCO	SNJACIC1	TUMANC1	TELEFBC1	TEF	SPCCPI1	SIDERC1	EGEPIBC1	PODERC1	SAGAC1	HIDROSII	GBVLAC1	
2/01/2012	15.15	23	101.64	48	5.51	109.25	56.8	0.28	118.5	24	0.08	0.48	1.4	30	8	2.69	2.43	17.01	60.3	0.89	5	2.78	4.25	0.6	13.5	
3/01/2012	15.2	23	101.64	48.05	5.66	109.25	56.8	0.28	118.5	24	0.085	0.49	1.4	30	8	2.69	2.45	17.01	60.3	0.89	5	2.78	4.25	0.6	13.5	
4/01/2012	15.3	23	105	48.05	5.8	109	56.8	0.28	118.5	24	0.085	0.5	1.4	31.7	8	2.65	2.45	17.01	60.3	0.9	5	2.78	4.25	0.6	13.5	
5/01/2012	15.2	23	104.85	48.05	5.8	108.5	56.8	0.28	118.5	24	0.086	0.5	1.4	31.53	8	2.6	2.47	17	60.3	0.9	5	2.78	4.25	0.6	13.5	
6/01/2012	15.2	23	104.85	48	5.85	108.8	56.8	0.28	118.5	24	0.09	0.49	1.4	31.35	7.8	2.6	2.47	17	60.3	0.88	5	2.78	4.25	0.6	13.5	
9/01/2012	15.2	23	104.85	48	5.7	105.4	56.8	0.28	115.5	24	0.09	0.49	1.4	31	7.8	2.6	2.47	16.7	60.3	0.85	5	2.78	4.25	0.6	13.5	
10/01/2012	15.2	23	104.85	48	5.7	104.6	56.4	0.28	115.5	24	0.09	0.49	1.4	30.95	7.8	2.69	2.52	16.55	60.3	0.86	5	2.78	4.25	0.6	13.5	
11/01/2012	15.4	23	107.65	48	5.8	109	56.4	0.28	115.5	24	0.09	0.49	1.4	31.95	8.1	2.7	2.55	17.21	60.3	0.88	5	2.78	4.25	0.6	13.5	
12/01/2012	15.4	22.75	107.65	49	5.8	109	56.4	0.28	116	24	0.09	0.49	1.4	32.3	8.1	2.61	2.6	17	60.3	0.86	5	2.78	4.25	0.6	13.5	
13/01/2012	15.4	24	107.65	49	5.9	108.08	56.4	0.28	116	24	0.09	0.51	1.4	33.3	8.1	2.68	2.55	17.35	61	0.89	5.0276	2.78	4.25	0.6	13.5	
16/01/2012	15.49	24	106.7	49	5.83	108	60	0.28	116	24	0.09	0.5	1.4	33	8.1	2.65	2.55	16.99	60.01	0.89	5.0552	2.78	4.25	0.6	13.5	
.
.
.
2/12/2022	7.81	34.6	21.65	6.06	1.54	151.7	198.5	1.68	185	18.5	0.05	0.39	0.72	61.36	7	0.98	0.68	3.7	58.52	1.56	6.8	9	8.97	0.22	2.39	
5/12/2022	7.85	34.6	30	6.14	1.56	149.8	198.5	1.68	185	18.5	0.05	0.393	0.72	61.97	7	0.98	0.68	3.69	58.52	1.56	6.8	9.15	8.97	0.22	2.34	
6/12/2022	7.75	34.6	30	6.14	1.57	147.93	198.5	1.6	185	18.5	0.05	0.391	0.72	61.97	7	0.97	0.68	3.62	58.52	1.56	6.8	9.15	8.97	0.22	2.34	
7/12/2022	7.75	34.6	30	6.14	1.57	147.7	198.5	1.6	185	18.5	0.05	0.39	0.72	61.3	7	0.97	0.68	3.62	58.52	1.58	6.8	9.15	8.97	0.22	2.34	
12/12/2022	7.5	34.6	29.1	6.14	1.6	144.16	198.5	1.67	185	18.5	0.05	0.395	0.72	59.71	7	0.97	0.68	3.62	58.52	1.6	6.8	9.15	8.97	0.22	2.34	
13/12/2022	7.5	34.6	29.1	6.12	1.62	140	198.5	1.67	185	18.5	0.05	0.395	0.72	60.08	7	0.92	0.68	3.46	58.52	1.59	6.8	9.15	8.97	0.22	2.34	
14/12/2022	7.5	34.6	29.9	6.12	1.61	138.7	198.5	1.67	185	18.5	0.05	0.361	0.72	61.12	7	0.92	0.68	3.46	58.52	1.6	6.8	9.15	8.97	0.22	2.34	
15/12/2022	7.4	34.6	29.9	6.12	1.6	137.41	198.5	1.67	185	18.5	0.05	0.362	0.72	61	7	0.92	0.68	3.46	58.52	1.6	6.8	9.15	8.97	0.22	2.34	
16/12/2022	7.4	34.6	29.9	6.12	1.58	133.9	198.5	1.67	185	18.5	0.05	0.362	0.72	61	7	0.92	0.68	3.46	58.52	1.6	6.8	9.15	8.97	0.22	2.34	
19/12/2022	7.5	34.6	29.9	6.12	1.59	133.7	198.5	1.67	185	18.5	0.05	0.353	0.72	61	7	0.92	0.68	3.38	58.52	1.65	6.8	9.15	8.97	0.22	2.34	
20/12/2022	7.5	35	29.9	6.12	1.59	134.05	198.5	1.67	185	18.5	0.05	0.346	0.72	61	7	0.92	0.68	3.38	58.1	1.63	6.8	9.15	8.97	0.22	2.34	
21/12/2022	7.8	35	29.9	6.12	1.6	136	198.5	1.67	184	18.5	0.05	0.354	0.72	60.5	7	0.91	0.68	3.38	58	1.61	6.8	9.15	8.97	0.22	2.34	
22/12/2022	7.8	35	29.9	6.12	1.63	135.7	198.5	1.67	184	18.5	0.05	0.355	0.72	61	7	0.89	0.68	3.34	58	1.61	6.8	9.15	8.97	0.22	2.34	
23/12/2022	8	35	29.9	6.12	1.65	133.14	198.5	1.67	184	18.5	0.05	0.355	0.72	60	7	0.89	0.68	3.42	58	1.61	6.8	9.15	8.97	0.22	2.34	
26/12/2022	8.01	35	29.9	6.12	1.65	134.75	198.5	1.67	184	18.5	0.05	0.36	0.72	61	7	0.89	0.68	3.42	58	1.62	6.8	9.15	8.97	0.22	2.34	
27/12/2022	8	35	29.9	6.12	1.65	134.75	198.5	1.67	184	18.5	0.05	0.36	0.72	61	7	0.89	0.68	3.42	58	1.61	6.8	9.15	8.97	0.22	2.42	
28/12/2022	8	35	29.9	6.12	1.7	133.75	198.5	1.67	184	18.5	0.05	0.36	0.72	61.99	7	0.89	0.45	3.42	58	1.61	6.8	9.4	8.97	0.22	2.42	
29/12/2022	8	35	27.9	6.12	1.7	134.7	198.5	1.67	184	18.5	0.05	0.36	0.72	61.5	7	0.89	0.45	3.42	58	1.63	6.8	9.4	8.97	0.22	2.45	
30/12/2022	7.85	35.1	27.9	6.12	1.7	135.55	198.5	1.67	184	18.5	0.05	0.36	0.72	61.57	7	0.89	0.45	3.42	58	1.64	6.8	9.4	8.97	0.22	2.6	
2/01/2023	7.9	35.1	27.9	6.12	1.69	135.25	198.5	1.67	184	18.5	0.05	0.36	0.72	61.57	7	0.89	0.45	3.42	58	1.64	6.8	9.4	8.97	0.22	2.6	

Anexo 4

Lista de empresas seleccionadas para procesar en algoritmo genético

Empresa	Último dato del set de prueba	Predicción del día 21 siguiente	Cumple con la restricción
CASAGRC1	7.9	8.52	Si cumple
BUENAVC1	27.9	28.28	Si cumple
BBVAC1	1.69	1.80	Si cumple
ATCU	0.115	0.13	Si cumple
FALABEC1	0.5	1.04	Si cumple
ELCOMEI1	0.7	0.86	Si cumple
CVERDEC1	29.8	36.35	Si cumple
CREDITC1	2.6	2.78	Si cumple
CPACASC1	3.95	5.02	Si cumple
CORAREC1	1.92	1.96	Si cumple
LUISAI1	184	222.13	Si cumple
INVCENC1	1.4	2.09	Si cumple
INTERBC1	2.6	2.76	Si cumple
IFS	23.9	25.74	Si cumple
SCOTIAC1	17	22.57	Si cumple
PML	0.091	0.11	Si cumple
MINSURI1	3.55	3.77	Si cumple
MIRL	0.054	0.07	Si cumple
NEXAPEC1	2	3.54	Si cumple
PERUBAI1	3.79	3.90	Si cumple
POSITIC1	0.72	0.86	Si cumple
TELEFBC1	0.45	0.64	Si cumple
TEF	3.42	4.08	Si cumple
SAGAC1	8.97	9.36	Si cumple
	Total		24