Universidad Nacional Federico Villarreal

VRIN | VICERRECTORADO DE INVESTIGACIÓN

ESCUELA UNIVERSITARIA DE POSGRADO

IMPLEMENTACIÓN DE UN MODELO ALGORÍTMICO DE INTELIGENCIA ARTIFICIAL PREDICTIVA - HILL CLIMBING PARA LA PREDICCIÓN DE RECAUDACIÓN TRIBUTARIA DEL ESTADO PERUANO

Línea de investigación:
Sistemas inteligentes, robótica, domótica

Tesis para optar el Grado Académico de Doctor en Ingeniería de Sistemas

Autor

Arpasi Chura, Rodolfo Fredy

Asesor

Herrera Salazar, José Luis

ORCID: 0000-0002-8869-3854

Jurado

Cachay Boza, Orestes

Rojas Romero, Karin Corina

Carrillo Balceda, Jesús Elías

Lima - Perú

2025

IMPLEMENTACIÓN DE UN MODELO ALGORÍTMICO DE INTELIGENCIA ARTIFICIAL PREDICTIVA - HILL CLIMBING PARA LA PREDICCIÓN DE RECAUDACIÓN TRIBUTARIA DEL ESTADO PERUANO

INFORME DE ORIGINALIDAD

25%	23%	5%	10%
INDICE DE SIMILITUD	FUENTES DE INTERNET	PUBLICACIONES	TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	qdoc.tips Fuente de Internet	3%
2	www.scielo.org.bo Fuente de Internet	2%
3	dspace.espoch.edu.ec Fuente de Internet	1%
4	hdl.handle.net Fuente de Internet	1%
5	www.coursehero.com Fuente de Internet	1%
6	www.scielo.org.co Fuente de Internet	1%
7	bibliotecas.ucasal.edu.ar Fuente de Internet	1%
8	repositorio.unfv.edu.pe Fuente de Internet	1%

Universidad Nacional Federico Villarreal

VRIN | VICERRECTORADO DE INVESTIGACIÓN

ESCUELA UNIVERSITARIA DE POSGRADO

IMPLEMENTACIÓN DE UN MODELO ALGORÍTMICO DE INTELIGENCIA ARTIFICIAL PREDICTIVA - HILL CLIMBING PARA LA PREDICCIÓN DE RECAUDACIÓN TRIBUTARIA DEL ESTADO PERUANO

Línea de investigación:

Sistemas inteligentes, robótica, domótica

Tesis para optar el Grado Académico de Doctor en Ingeniería de Sistemas

Autor

Arpasi Chura, Rodolfo Fredy

Asesor

Herrera Salazar, José Luis

ORCID: 0000-0002-8869-3854

Jurado

Cachay Boza, Orestes

Rojas Romero, Karin Corina

Carrillo Balceda, Jesús Elías

Lima - Perú

2025

DEDICATORIA

Dedico este trabajo de investigación con mucho afecto a mis queridos padres Adolfo y Vilma, a mi amada esposa Danya por su invalorable apoyo y a mis preciados hijos Ian, Raisa y Thiago por su amor infinito.

AGRADECIMIENTOS

A la Escuela Universitaria de Posgrado de la UNFV por haberme acogido en sus aulas y darme la oportunidad de realizar mi trabajo de investigación.

A los doctores: Orestes Cachay Boza, Karin Corina Rojas Romero, Jesús Elías Carrillo Balceda por sus sugerencias que han contribuido en el desarrollo de la tesis.

Al Dr. José Luis Herrera Salazar y al Magíster en Ciencias Wilfredo Mamani Ticona, por su asesoría en la tesis.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE DE ECUACIONES

ABREVIATURAS

AUTO-ARIMA: *Autoregressive Integrated Moving Average*

ABE : *Attribute-Based Encryption*

CAMF : *Cellular Automata-based heuristic for Minimizing Flow*

GBR : *Gradient Boosting Regressor*

IA *: Inteligencia Artificial*

KNNR : *K-Nearest Neighbors Regressor*

MAPE : *Mean Absolute Percentage Error*

MLP-O : *Multi-Layer Perceptron* con optimización

MLPR : *Multi-Layer Perceptron Regressor*

PLL : *Phase Locked Loop*

RFR : *Random Forest Regressor*

SUNAT : Superintendencia Nacional de Aduanas y de Administración Tributaria

RESUMEN

Determinar el mejor modelo de predicción de recaudación tributaria del Estado peruano, con la implementación de la Red Neuronal y el algoritmo *Hill Climbing* contrastado con otros modelos de aprendizaje supervisados, como *MLPR, GBR, RFR, KNNR* y el estadístico AutoARIMA. La metodología se desarrolló en cinco fases: obtención y reprocesamiento de la serie temporal, construcción de las variables de entrada, identificación de la estructura MLP óptima en el período de validación, predicción en el período de prueba. Los resultados obtenidos de la métrica MAPE fueron: AutoARIMA (3.0691 %), MLP-O (3.3830 %), *MLPR* (3.5923 %), *GBR* (3.6660 %), *RFR* (5.3389 %), *KNNR* (6.4466 %), donde claramente el resultado MAPE del modelo MLPR con el modelo (MLP-O) se muestra superior. Esto ocurrió con la inclusión del algoritmo *Hill Climbing*. Se puede verificar que los modelos *MLPR* y *GBR* presentan resultados similares, pero no mejores a los resultados de los modelos AutoARIMA y MLP-O. El poder de predicción del algoritmo propuesto en la presente investigación, basado en un modelo híbrido de redes MLP y algoritmo *Hill Climbing* con datos de la serie temporal de la SUNAT de los años 2000 a 2023, fue eficaz. Se obtuvo un error MAPE de 3.383 %. La predicción de la recaudación del año 2023 fue de 192 696 238 616.60, y el valor recaudado del mismo año fue de 192 128 881 908.62, lo que demuestra la eficacia del algoritmo.

Palabras clave: Hill Climbing, hyperparameter tuning, recaudación tributaria.

ABSTRACT

This study aims to determine the most accurate predictive model for tax revenue collection by the Peruvian government through the implementation of a Neural Network enhanced with the Hill Climbing algorithm, compared against other supervised learning models such as MLPR, GBR, RFR, KNNR, and the statistical AutoARIMA model. The methodology was developed in five stages: acquisition and preprocessing of the time series data, construction of input variables, identification of the optimal MLP structure during the validation phase, and forecasting during the testing phase. The Mean Absolute Percentage Error (MAPE) results were as follows: AutoARIMA (3.0691%), MLP-O (3.3830%), MLPR (3.5923%), GBR (3.6660%), RFR (5.3389%), and KNNR (6.4466%). Notably, the MAPE results obtained from the MLPR model combined with the optimized MLP (MLP-O) model proved superior, particularly due to the integration of the Hill Climbing algorithm. While MLPR and GBR yielded similar results, they did not surpass the performance of AutoARIMA and MLP-O. The predictive capability of the proposed hybrid model, combining MLP networks with the Hill Climbing algorithm using SUNAT's time series data from 2000 to 2023, was effective. A MAPE of 3.383% was achieved. The predicted tax revenue for 2023 was 192 696 238 616.60, while the actual revenue collected was 192 128 881 908.62, demonstrating the algorithm's accuracy.

Keywords: Hill Climbing, hyperparameter tuning, tax revenue forecasting.

I. INTRODUCCIÓN

1.1. Planteamiento del problema

Para Night & Bananuka (2019), Celikay (2020) y Flores (2022) los impuestos son la principal fuente de ingresos del Estado, a pesar de que muchos países han implementado estrategias para la recaudación tributaria, aun así, el incumplimiento tributario persiste debido a la evasión fiscal que es bastante común, como la reducción de los montos de los impuestos adeudados a través de empresas ficticias. Así mismo, el desarrollo de una nación se ve afectado en la medida en que sus ciudadanos cumplen voluntariamente con sus obligaciones tributarias, por tanto, el gobierno recauda los fondos económicos de los contribuyentes y conocer cuanto recaudarán sería aún mejor, para que puedan planificar la distribución de la misma con responsabilidad social.

Asimismo, según Ramírez (2020), Santiago et al., (2017), Arciniegas et al. (2021) la recaudación de tributos sigue siendo un problema para la administración tributaria, por lo que constantemente generan normativas para un mejor control y mejor incentivo a los contribuyentes, también se han venido desarrollando modelos y herramientas estadísticas, en los que destacan el ARIMA se considera una herramienta predictiva, su flexibilidad en la incorporación de herramientas estadísticas, econométricas y matemáticas para estimaciones de valores recaudados de una serie de tiempo con un margen de error aceptable para los pronósticos tributarios, en otro caso, la metodología de *Box Jenkins* posee procedimientos para identificar, ajustar y verificar los modelos ARIMA con los datos de la serie de tiempo.

Arciniegas Paspuel et al. (2021) sostiene que la recaudación tributaria en el Ecuador de los años 2016 a 2020 con presencia de la COVID-19, tuvo un impacto negativo en las recaudaciones tributarias por efecto de la pandemia, sin embargo, para un análisis que garantice el pronóstico con mayor precisión la recaudación tributaria usaron herramientas estadísticas de series de tiempo en los pronósticos como el modelo ARIMA con un MAPE de 1.52 %

considerado como el más idóneo, con rangos aceptables de error, esto permitió mejorar la estimación de los ingresos fiscales.

En la presente investigación se logró identificar el mejor modelo de predicción de la recaudación tributaria del Estado peruano para el período enero a diciembre de 2023 con la implementación de la red neuronal *MLP* y el algoritmo *Hill Climbing*, comparado con otros modelos de aprendizaje supervisado *MLPR* (*Multilayer Perceptron Regressor*), GBR (*Gradient Boosting Regressor*), *RFR* (*Random Forest Regressor*), *KNNR* (K-*Nearest Neighbors Regressor*), y el método estadístico AutoARIMA (*Auto-Regressive Integrated Moving Average*).

1.2. Descripción del problema

En la literatura nacional e internacional sobre recaudación tributaria, se menciona que existe una serie de estrategias implementadas, como la cultura tributaria, incentivos, exenciones, campañas de concientización, educación tributaria entre otros, con la única finalidad de que el Estado pueda recaudar los tributos de manera sostenible siendo un tema de importancia de modo que pueda asumir sus obligaciones y con el desarrollo del país, e inclusive en momentos de incertidumbre como los riesgos biológicos como la pandemia COVID-19 que influyó en el crecimiento económico y reducción de la recaudación tributaria (Moreno Kong, 2018).

Herrera Maguiña (2021) junto al programa Centros de Asistencia Financiera y Tributaria (NAF) se proponen iniciativas en el Perú para generar un sentido de responsabilidad económica en la población siendo esta de mucha importancia en la cultura tributaria y los incentivos que otorga el Estado a sus ciudadanos contribuyentes, por otro lado, la SUNAT brinda iniciativas complementarias sobre el valor de las exenciones tributarias como la devolución del Impuesto General a las Ventas (IGV) y la contribución del Impuesto de

Promoción Municipal (IPM) para llevar a cabo actividades nacionales e internacionales, operaciones, proyectos de desarrollo social, los gobiernos suelen utilizar exenciones fiscales para impulsar los ingresos fiscales a corto plazo, sin embargo, la no aplicación de manera justa, las exenciones fiscales pueden socavar la justicia subyacente del sistema fiscal (Kaldor, 2021; Lauletta & Montaño Campos, 2018).

Esto se complementa con la cultura tributaria el cual tiene impacto significativo en la recaudación de ingresos de los países y como herramienta para reducir la evasión fiscal de los contribuyentes del impuesto a la renta de tercera categoría, el fomento de la recaudación de impuestos a través de campañas de concientización, la educación financiera tiene una influencia significativa en la recaudación de impuestos (Pérez Valqui, 2018; Cabrera et al., 2021; Piancastelli & Thirlwall, 2020; Romero-Carazas et al., 2022)

También, es importante que se establezcan impuestos justos sin que esta afecte el dinamismo de la economía y de la productividad nacional y que los ciudadanos puedan tomar un mayor control de su economía (Coaquira Taboada et al., 2022).

En este sentido se vuelve importante conocer el modelo de predicción óptimo para la recaudación tributaria del Estado peruano con la implementación de una Red Neuronal Artificial con el algorítmico de búsqueda aleatoria - *Hill Climbing* para la predicción de recaudación tributaria del Estado peruano, comparado con otros modelos de aprendizaje inteligentes, siendo ésta un estímulo para el desarrollo de la infraestructura tecnológica en el Perú.

1.3. Formulación del problema

1.3.1. Problema general

¿Cuál es el modelo algorítmico de mejor predicción de inteligencia artificial *MLP – Hill Climbing* para la recaudación tributaria del Estado peruano comparado con otros modelos de aprendizaje supervisado y el método estadístico?

1.3.2. Problemas específicos

A. ¿Cómo obtener la serie temporal de datos de la recaudación tributaria de la SUNAT para su procesamiento y construcción de las variables de entrada desde los años 2000 a 2023?
B. ¿Cómo elaborar las estructuras de procesamiento de los algoritmos MLP óptimo con *Hill Climbing*, MLPR, GBR, RFR, KNNR, AutoARIMA en el periodo de validación?
C. ¿Cómo evaluar el modelo entrenado en el período de prueba para una mejor estimación de la recaudación tributaria?
D. ¿Cuánto difieren estadísticamente los porcentajes de promedio de error de los algoritmos en la predicción de recaudación tributaria?

1.4. Antecedentes

Aplicación de Algoritmos Híbridos

Lajunen (2014) la aplicación de algoritmos híbridos en las simulaciones de rutas operativas reales que se habían medido a partir de rutas populares de camiones en el sur de Finlandia, las variables de estudio fue el consumo de combustible por tonelada-kilómetro de carga útil, disminuyó una media del 17 % cuando el peso total de la combinación aumentó de 40 t a 60 t. La disminución es del 23 % cuando se pasa de 40 t a 76 t y del 28 % cuando se pasa de 40 t a 90 t. Según los resultados de la simulación, el ahorro de combustible de una combinación de vehículos pesados puede mejorarse hasta un 6 % mediante algoritmos híbridos.

Booba et al. (2024) implementaron el Algoritmo de Búsqueda de enfoque combinado *Generalized Backtracking Regularized Adaptive Matching Pursuit and Adaptive* (GBRAMP) *β-Hill Climbing Algorithm for Virtual Machine Allocation in Cloud Computing* (BA-VMA-CC), se utilizó para el proceso de migración de máquinas virtuales (VM) y Adaptive β-Hill *Climbing Algorithm* se aplicó en la colocación de máquinas virtuales. Estas dos tareas son elementos esenciales de la asignación de máquinas virtuales GBRAMP se utilizó para minimizar el coste y la energía tanto para los proveedores de servicios en la nube como para los usuarios con la ayuda del proceso de migración y para ahorrar tiempo y energía. El Algoritmo Adaptativo *β-Hill Climbing* (AβHCA) se empleó para maximizar la eficiencia, minimizar el consumo de energía y el desperdicio de recursos. Combinando ambos GBRAMPA-AβHCA VM se asigna óptimamente en PM (*Physical Machines*) con alta eficiencia minimizando el coste y el consumo de energía. El resultado, es la implementación con éxito del enfoque combinado de GBRAMP y Adaptive β-*Hill Climbing* para la asignación de máquinas virtuales en *Cloud Computing*. Se consiguió un mejor equilibrio entre los factores de exploración y explotación en el espacio de búsqueda mediante la combinación de dos algoritmos. Así, los objetivos del problema de asignación de máquinas virtuales se optimizaron de forma significativa cuando se combinan estos dos procedimientos.

Ioniţă et al. (2023) implementaron el algoritmo heurístico de la versión *Hill Climbing* no añadió ninguna sobrecarga de tiempo adicional en el proceso por el contrario se ha mejorado el rendimiento de los esquemas de Cifrado Basado en Atributos *(ABE - Attribute-Based Encryption),* para circuitos booleanos, por ello, se muestra la mejora los sistemas ABE para circuitos booleanos. Se propuso el método para optimizar los circuitos booleanos monótonos reescribiéndolos en una forma más simple y equivalente. Cada una de las heurísticas se comportó de manera diferente en términos de optimización y tiempo de ejecución, dependiendo del tamaño y la estructura del circuito booleano.

Aplicación de Algoritmos *Hill Climbing*

Castillo-Reyes et al. (2024) aplicaron algoritmos de ajuste de metaparámetros en procesos de erosión del suelo y transporte de sedimentos, la heurística CAMF *(Cellular Automata-based heuristic for Minimizing Flow)* selecciona los lugares de repoblación forestal para minimizar la afluencia de sedimentos a la salida de una cuenca. CAMF utilizó una representación ráster de la cuenca y una heurística de optimización *Hill – Climbing*. El tiempo de ejecución puede no ser el adecuado para grandes conjuntos de datos, sin embargo, el método de optimización pudo reducir el número y el coste de las iteraciones. La eficacia de las intervenciones destinadas a minimizar los daños en los lugares seleccionados dentro de una cuenca hidrológica es muy específica de cada lugar, debido a la variabilidad espacial de la intensidad de la erosión del suelo y de la interacción espacial en los procesos de transporte de sedimentos. El método CAMF seleccionó una representación ráster de la cuenca, aquellas celdas en las que una intervención, como la repoblación forestal, minimizó la pérdida de sedimentos en la salida, bajo unas restricciones dadas. Este problema de optimización espacial se resuelve mediante el método *Hill -Climbing* que proporcionó una alternativa robusta y de convergencia rápida a los métodos exactos y heurísticos. Sin embargo, el coste computacional de CAMF aumenta sustancialmente con el incremento del tamaño del problema.

Wang et al. (2023) utilizaron el algoritmo *Hill-Climbing* para determinar la detección dinámica por Espectroscopia Raman Mejorada, se estableció un proceso de evaporación acelerada a alta temperatura para obtener un punto caliente, a continuación, se enfría rápidamente, utilizaron la intensidad espectral como señal de retroalimentación en el algoritmo *Hill Climbing* para mover la etapa hacia arriba y hacia abajo con el fin de ajustar la profundidad del láser en las muestras y realizar el enfoque automático. En este proceso, se utilizó la media de las tres primeras intensidades máximas de pico como función de evaluación del enfoque se

recorrió la posición máxima en todo el proceso con un paso grande, luego se da un paso atrás y después se desplaza con una distancia pequeña hasta encontrar el punto con la mejor intensidad espectral.

Zhang et al. (2023) propusieron el algoritmo *Hill - Climbing* para la captura de frecuencias y una estrategia de control difusa basada en la impedancia mínima para el seguimiento de frecuencias aplicada al taladro ultrasónico de percusión, identificando rápidamente la frecuencia de resonancia y lograron un taladrado rápido y estable. La estrategia de control propuesta redujo el tiempo de captura de frecuencia de 3.1 s a 1.8 s en comparación con el control de barrido de frecuencia utilizado habitualmente cuando se identifica la misma frecuencia. Para conseguir una perforación estable del taladro ultrasónico percusivo en poco tiempo, se estudiaron los algoritmos de control para la captura y seguimiento de la frecuencia resonante. Los resultados experimentales mostraron que el algoritmo *Hill – Climbing* propuesto para la captura de frecuencias redujo el tiempo en 1.3 s en comparación con el escaneo de frecuencias, y el control difuso propuesto para el seguimiento de frecuencias reduce la impedancia mínima en 19.5 Ω y el ángulo de fase en 9.7° en comparación con el seguimiento PLL *(Phase Locked Loop).*

Naskar et al. (2023) aplicaron los algoritmos de búsqueda armónica (HS) inspirado en la música, integrada con el método *Late Acceptance Hill Climbing* (LAHC), para una mejor explotación, el método propuesto se ha probado en varios conjuntos de datos UCI *Machine Learning Repository* de la Universidad de California, junto con un conjunto de datos de COVID-19 y conjuntos de datos de expresión génica basados en microarrays utilizados para la identificación de genes/biomarcadores del cáncer, los resultados mostraron un mejor rendimiento en comparación con los algoritmos metaheurísticos de selección de características más avanzados, se analizaron los resultados mediante sensibilidad, convergencia y boxplot para profundizar en el comportamiento del algoritmo, sin embargo, el método de búsqueda local

aumentó el tiempo medio de entrenamiento, lo que puede solucionarse en el futuro con mejores estrategias dinámicas de exploración y explotación.

Bhattacharya et al. (2023) aplicaron algoritmos híbridos de optimización metaheurístico híbrido AdBet-WOA (*Whale Optimization Algorithm With Integrated Adaptive -Hill Climbing local search*) y el clasificador *Support Vector Machine (SVM)* clasifican los datos de prueba de cáncer de colon, pulmón y ambos combinados con una precisión del 99.99% lo que permitió lograr mayor eficiencia computacional y la clasificación de imágenes con precisión utilizando el menor número de características extraídas.

Al-Betar et al. (2023) implementaron el método de búsqueda local llamado *Hill – Climbing* que mejoró hibridándolo con las técnicas de optimización ELD *(Economic Load Dispatch)* ya que es rígido y profundo, y HHO *(Harris Hawks Optimizer)* mejorando significativamente, el HHO puede navegar por varias regiones potenciales del espacio de búsqueda, mientras que *Hill Climbing* se utilizó para buscar en profundidad la solución óptima local en cada región potencial el tiempo de ejecución del método híbrido es mayor que el método original.

Aplicación de las Redes Neuronales en las Finanzas

Para Del Carpio Gallegos (2005) la aplicación de las Redes Neuronales en el tema financiero es amplio en especial, en el pronóstico de variables financieras, la mayoría de los estudios tienen que ver con la predicción de rendimientos de valores, quiebras, que en general estas herramientas inteligentes ayudan a tomar mejores decisiones financieras, también, son aplicados como herramientas complementarias a los enfoques tradicionales de análisis multivariante.

Molina-Muñoz (2021) el uso de *Machine Learning* en finanzas se ha convertido en un *"hot topic"* dentro de la literatura financiera, por los modernos algoritmos computacionales

que han permitido la estimación de modelos, desafiantes anteriormente, con niveles de precisión muy satisfactorios, como la aplicación de metodologías *K-means,* que es un algoritmo de aprendizaje no supervisado utilizado para agrupamiento de datos *(clustering),* se divide en un conjunto de datos en K grupos basándose en la similitud entre ellos.

Ludeña Dávila & Tonon Ordóñez (2021) el cálculo del riesgo de insolvencia se ha convertido en un parámetro importante, buscando anticiparse a la eventualidad de llegar a tener un problema económico y generar insolvencia, las redes neuronales facilita el cálculo del riesgo de insolvencia, generan resultados más efectivos que los métodos tradicionales.

Rodriguez-Tovar et al. (2023) en el análisis de prevención de fraudes empresariales en el sistema financiero fue efectivo con el uso de técnicas de aprendizaje automático, con aplicación de algoritmos basados en árboles de decisión *C5.0-SVM (Support Vector Machine con C5.0), Naïve Bayes y Random Forest*, sin embargo, según literatura se mejoró los resultados con aplicaciones de algoritmos basados en aprendizaje profundo, realizando la correcta clasificación de las neuronas *iNALU (Improved Neural Arithmetic Logic Unit) y ReLU (Rectified Linear Unit)* el porcentaje de efectividad incrementó en gran proporción.

Hernández Aros et al. (2023) para la solución de problemas que afectan las finanzas de las organizaciones, disminución del valor económico que afecta la dinámica empresarial y la prevención de fraudes empresariales, el uso de técnicas de aprendizaje automático y profundo permitió generar prevención, tratamiento y resolución a los fraudes en el sistema financiero, para este fin, se aplicaron los algoritmos de árboles de decisión, *C5.0-SVM, Naïve Bayes y Random Forest,* con porcentajes de: 92 %, 83.15 %, 80.4 % y 76.7 %.

Gutierrez Portela et al. (2023) muchas organizaciones en la actualidad se ven afectadas por fraudes financieros perjudicando directamente el patrimonio de las empresas, para ello se han implementado técnicas supervisadas y no supervisadas que usan inteligencia artificial para

la prevención y detección de estos fraudes y así minimizar riesgos en las operaciones financieras, utilizan algoritmos basados en árboles de decisión, *Naive Bayes*, Máquina de vectores de soportes, Bosque aleatorio y Regresión logística.

Políticas fiscales en Latinoamérica

Caro Arroyo (2020) los modelos de tributación desarrollados en la región latinoamericana, tienen un enfoque proteccionista para la acumulación del capital, por ejemplo, en estos casos, el gasto social y el funcionamiento del Estado es sobrellevado en su mayoría por los impuestos que gravan al consumo, bajo el supuesto de brindar al capital posibilidades de seguir incrementándose en los mercados financieros, por otro lado, las estructuras tributarias no solo son burocráticas por las formas como se regula, sino también, existe un ambiente de desconfianza, descontento por las crisis económicas que finalmente son pagadas por las clases más bajas de la sociedad.

Desfrancois (2023) para una buena gobernanza pública una característica clave es la transparencia fiscal, es fundamental para una gestión fiscal y una rendición de cuentas eficaces con exhaustividad, claridad, confiabilidad.

Morales Millan & Cely García (2024) las políticas tributarias ineficaces, sistemas tributarios complicados, informalidad económica, escasa cultura tributaria, corrupción, debilidad institucional, omisión de declaraciones de ingresos, manipulación de documentos, son consideradas como causas que permiten la evasión fiscal, lo que constituye de por sí en un fenómeno complejo, las consecuencias, pueden ser, la disminución de ingresos fiscales, aumento de la desigualdad, deterioro de los servicios públicos, entre otros, esta situación tiene un efecto negativo en la capacidad de inversión en proyectos.

1.5. Justificación de la investigación

Para la gestión gubernamental, las predicciones de recaudación son importantes porque permite un mejor Planeamiento Estratégico de la Superintendencia Nacional de Aduanas y de Administración Tributaria (SUNAT), de modo que le permita lograr sus objetivos como, por ejemplo, conocer los importes de recaudación de Ingresos Fiscales, y mejorar el cumplimiento tributario y aduanero, frente a la recaudación tributaria es bastante complicado en el Perú, porque la cultura tributaria no es una fortaleza ya que la informalidad en la Economía peruana está incrementándose cada vez más hasta en un 78 % este año (ESAN, 2024), por otro lado, los incentivos, las campañas de concientización, la educación tributaria son casi nulos como estrategia del gobierno de turno para recaudar los tributos.

Como Línea de Investigación, las redes neuronales representan una línea de investigación muy poco desarrollado en el país, consecuentemente, es una oportunidad que presenta la Escuela Universitaria de Posgrado a sus egresados de desarrollar ésta línea de investigación, la potencialidad de las redes neuronales en la predicción de series temporales es de sumo interés, que pronostique valores futuros de la recaudación tributaria peruana y que permita mostrar su precisión respecto a técnicas estadísticas convencionales, sus métodos algorítmicos ha permitido comprender fenómenos complejos modelando a través de sistemas informáticos los sistemas naturales.

Para el Desarrollo Tecnológico, las redes neuronales tienen múltiples aplicaciones a diversas áreas del conocimiento: salud, ingeniería, educación, gestión gubernamental, etc., esta tecnología acompañará a la humanidad en los siguientes años, permitiendo la automatización de tareas complejas al grado que las redes neuronales tienen la capacidad de aprender y adaptarse a partir de datos, lo que las hace ideal para tareas de predicción, clasificación, reconocimiento de patrones y optimización en una amplia gama de campos, facilitando una

innovación constante en el reconocimiento de imágenes, análisis de textos, voz, procesamiento de grandes bancos de información.

Finalmente, el aporte del trabajo de investigación es encontrar el mejor modelo de predicción inteligente de la red MLP y algoritmo *Hill Climbing* a través de la metodología propuesta en la Figura 3, de esta forma realizar las predicciones de la recaudación tributaria del Estado peruano con mayor eficiencia contrastado con otros modelos inteligentes como *GBR, KNNR, MLPR, RFR* y con el modelo estadístico AutoARIMA, esto con la finalidad de aportar en el desarrollo de la infraestructura tecnológica en el Perú de forma académica.

1.6. Limitaciones de la Investigación

El acceso a la información pública de la serie temporal fue extraída del *website* de la SUNAT, la serie utilizada corresponde a la suma de los impuestos Ingresos Tributarios recaudados por la SUNAT – Internos; Ingresos Tributarios recaudados por la SUNAT – Aduaneros; Contribuciones Sociales; Ingresos No Tributarios, de los años 2000 a 2023, sin embargo, el acceso a la información del año 2000 a 2005 no está disponible en la página web de SUNAT, por lo que fue necesario realizar las solicitudes respectivas (SUNAT, 2024).

Desde el punto de vista tecnológico se requiere de plataforma de hardware de alta performance para la ejecución del programa informático, que no está disponible para estudiantes e investigadores en la universidad, por lo que fue necesario alquilar servidores de Google, por otro lado, la conectividad a internet es una limitante ya que el Perú aún no ha integrado la plataforma de comunicación de datos en las instituciones educativas tanto a nivel de educación básica regular como superior, más aún, esto limita el procesamiento de grandes volúmenes de información necesarios para el entrenamiento de los algoritmos de inteligencia artificial.

Comexperu (2024), Espinoza et al. (2022), Instituto Peruano de Economía (2020) la economía del Perú se contrajo en 30 % y 9 %, en el año 2020 como efecto directo de la Pandemia que paralizó las actividades económicas afectando la economía nacional siendo el 71 % de informalidad, la caída de la demanda provocó el cierre de puestos de trabajo, incremento de competencia desleal, incremento de gastos operativos entre otros que impactaron la economía nacional, por otro lado, el impacto de economía tuvo el mismo impacto en las tecnologías de información y comunicación, aunque posteriormente la demanda por soluciones digitales se incrementó paulatinamente con el tema del teletrabajo, comercio electrónico, convirtiéndose en un desafío para la infraestructura digital del país, la brecha digital se incrementó durante la pandemia siendo esto en la actualidad una oportunidad emergente para el desarrollo de tecnología inteligente en el Perú.

1.7. Objetivos

1.7.1. Objetivo general

Determinar el mejor modelo de predicción de inteligencia artificial *MLP – Hill Climbing* para la recaudación tributaria del Estado peruano, comparado con otros modelos de aprendizaje supervisado *MLPR, GBR, RFR, KNNR* y el método estadístico AutoARIMA.

1.7.2. Objetivos específicos

A. Obtener la serie temporal de datos de la recaudación tributaria de la SUNAT para su procesamiento y construcción de las variables de entrada desde los años 2000 a 2023.
B. Elaborar las estructuras de procesamiento de los algoritmos *MPL* óptimo con *Hill Climbing*, *MLPR, GBR, RFR, KNNR*, AutoARIMA, para el período de validación.

C. Evaluar los modelos entrenados para el periodo de prueba.

D. Determinar la diferencia significativa de los porcentajes de promedio de error generados por los algoritmos en términos de eficiencia en la predicción.

1.8. Hipótesis

1.8.1. Hipótesis general

H_o: Si no es posible implementar el modelo algorítmico de inteligencia artificial *MLP– Hill Climbing* para la predicción de la recaudación tributaria del Estado peruano entonces no podemos afirmar que la predicción es mejor respecto de otros modelos de inteligencia artificial.

H_a: Si es posible implementar el modelo algorítmico de inteligencia artificial *MLP– Hill Climbing* para la predicción de la recaudación tributaria del Estado peruano entonces podemos afirmar que la predicción es mejor respecto de otros modelos de inteligencia artificial.

1.8.2. Hipótesis específicas

- H_o: No es posible obtener la serie temporal de datos de la recaudación tributaria de la SUNAT entonces no se procesa ni se construye las variables de entrada.

  H_a: Si es posible obtener la serie temporal de datos de la recaudación tributaria de la SUNAT entonces se procesa y construye las variables de entrada.

- Ho: No es viable elaborar las estructuras de procesamiento de los algoritmos MLP óptimo con Hill Climbing, *MLPR, GBR, RFR, KNNR*, AutoARIMA entonces no es factible comparar sus resultados para el periodo de validación.

  Ha: Si es viable elaborar las estructuras de procesamiento de los algoritmos MLP óptimo con Hill Climbing, *MLPR, GBR, RFR, KNNR*, AutoARIMA entonces es factible comparar sus resultados para el periodo de validación.

- Ho: No es posible evaluar el modelo entrenado en el período de prueba entonces no es posible generar las predicciones.

  Ha: Si es posible evaluar el modelo entrenado en el período de prueba entonces es posible generar las predicciones.

- H_o: No hay diferencia significativa entre los algoritmos en términos de eficiencia en la predicción.

  Ha: Existe al menos una diferencia significativa entre los algoritmos en términos de eficiencia en la predicción.

1.8.3. ***Hipótesis para probar la Normalidad***:

H_0: Los datos para los algoritmos provienen de una distribución normal

H_1: Los datos para los algoritmos no provienen de una distribución normal

II. MARCO TEÓRICO

2.1. Relación al marco filosófico

Hernández Márquez (2018) el algoritmo *Hill Climbing* conocido como ascenso de colina o de alpinismo es un algoritmo heurístico e iterativo que inicia dando una solución próxima al problema, luego el algoritmo realiza un cambio en algún elemento, si el resultado es una mejora en la solución entonces sucede un cambio incremental a la nueva solución, este procedimiento se repite hasta que ya no se puedan encontrar mejores soluciones, normalmente este resultado es un óptimo local pero no necesariamente garantiza la mejor solución. Otro aspecto importante a considerar es que el espacio de búsqueda se contrae en cada iteración para lograr esto se utiliza una regla determinista generalmente. Durante el proceso se desarrolla en dos etapas: la exploración y la aproximación, la primera etapa realiza las primeras iteraciones con la finalidad de buscar alguna solución, tratando de cubrir todo el espacio de búsqueda y finalmente cuando el espacio de búsqueda se hace más pequeño el algoritmo aumenta la precisión por aproximación.

Considerando estas características el marco filosófico más próximo estaría relacionado con el Reduccionismo, Viniegra Velázquez (2014) considera como una postura epistemológica que sostiene que el conocimiento de lo complejo debe ser, obligadamente reducida a sus componentes más simples, o que un sistema complejo solamente puede explicarse por la reducción hasta sus partes fundamentales, siendo necesario y suficiente para resolver los problemas del conocimiento.

Sin embargo, Fierro (2011) las Redes Neuronales estaría relacionado con el conexionismo, al igual que el Procesamiento Distribuido Paralelo, que hace un símil con los procesos cognitivos del hombre tanto en la rapidez con lo que se realizan estos procesos y su capacidad de resistir frente a daños, así mismo, la forma como operan

sus componentes para el almacenamiento de la información o el conocimiento de manera distribuida en patrones de activación según el nivel de complejidad conectiva que hayan desarrollado las redes neuronales. También, podríamos relacionar con una visión empirista conexionista, considerando que el conocimiento proviene de la experiencia.

2.2. Machine Learning

Es el estudio científico de algoritmos y modelos estadísticos que los sistemas de computación pueden usar para llevar a cabo una tarea específica sin usar instrucciones explícitas, sino más bien basándose en patrones e inferencias obtenidas de los datos.

2.3. Hyperparameter Tuning

Ghawi & Pfeffer (2019), Bergstra & Bengio (2012), Bergstra et al. (2011) en el aprendizaje automático, un hiperparámetro es un parámetro cuyo valor se fija al inicio del proceso de aprendizaje, para su optimización se elige un conjunto de hiperparámetros, lo que le permite encontrar una pareja de hiperparámetros que produzca un modelo óptimo que minimice una función de pérdida predefinida en datos independientes, los parámetros del modelo aprenden durante la fase de entrenamiento para ajustarse a los datos, en cambio, los hiperparámetros se establecen fuera del procedimiento de entrenamiento y se utilizarán en el control de ajuste de datos de entrenamiento y controlarán la flexibilidad del modelo al momento de adaptación de datos.

Es el proceso mediante el cual se busca el conjunto de hiperparámetros que logran maximizar el desempeño de un modelo a la hora de realizar predicciones, también, busca minimizar el tiempo y los recursos necesarios para lograr el primer objetivo, con la finalidad

de probar modelos de estimación, debido a que requieren muchos recursos computacionales y los modelos pueden tomar tiempo en converger, no solo en maximizar el desempeño sino también en un tiempo que sea plausible.

2.4. Forecasting Randomized Search

Syed et al. (2020) evaluaron el rendimiento de los modelos de predicción mediante aprendizaje automático distribuido. El ajuste de parámetros es prometedor, aunque supone un reto para optimizar el rendimiento. *Forecasting Randomized Search* es una técnica utilizada para la optimización de hiperparámetros en modelos de predicción de series temporales (*forecasting*). *Randomized Search*, método eficiente para encontrar los mejores parámetros de un modelo sin necesidad de probar todas las posibles combinaciones.

2.5. Gradient Boosting Regressor Trees (GBRT)

Zainab et al. (2020) este método tiende a mejorar la precisión gracias al hecho de que utilizaron un conjunto de datos de prueba independiente, ahorrando al mismo tiempo una cantidad considerable de tiempo de cálculo. *Gradient Boosting Regressor Trees* es un algoritmo basado en árboles de decisión que se usa para problemas de regresión, como la idea de *Boosting*, donde se entrenan y construyen múltiples árboles de forma secuencial, cada uno intenta corregir los errores del anterior se espera un ajuste adecuado de los parámetros para mostrar un rendimiento superior.

La librería *Scikit-learn*, tiene implementado una clase GBR, dentro de sus principales parámetros se puede mencionar:

- Control del modelo en general
    - *n_estimators:* Número de árboles en el ensamble.
    - *learning_rate:* Factor de reducción de la contribución de cada árbol.

- *loss:* Función de pérdida a minimizar.

- Control del crecimiento de los árboles
  - *max_depth:* Profundidad máxima de cada árbol.
  - *min_samples_split:* Número mínimo de muestras necesarias para dividir un nodo.
  - *min_samples_leaf:* Número mínimo de muestras en cada hoja de un árbol.
  - *max_features:* Número de características a considerar en cada división. Puede ser *auto*, *sqrt*, *log2* o un número específico.
  - *subsample:* Proporción de datos usados en cada iteración de entrenamiento. Un valor menor a 1 introduce aleatoriedad y ayuda a reducir el sobreajuste.
- Regularización y optimización
  - *alpha:* Parámetro para la pérdida *Huber*.
  - *ccp_alpha:* Parámetro de poda que controla la complejidad del árbol.
  - *max_leaf_nodes:* Número máximo de hojas en cada árbol, limitando su complejidad.
- Control de aleatoriedad
  - *random_state:* Fija la semilla para reproducibilidad.
  - *warm_start:* Si es True, reutiliza los árboles previos al ajustar más estimadores.

Los parámetros GBR se pueden ajustar mediante validación cruzada o con técnicas como *GridSearchCV* para encontrar la mejor configuración para el problema en cuestión.

2.6. Random Forests

Müller & Guido (2017) un bosque aleatorio es esencialmente una colección de árboles de decisión, donde cada árbol es diferente de los demás, cada árbol puede hacer un trabajo de

predicción relativamente bueno, pero es probable que sobreajuste parte de los datos. Si construimos muchos árboles, funcionarán bien y se ajustan de diferentes maneras, promediando sus resultados se reduce el sobreajuste, al tiempo que se mantiene el poder predictivo de los árboles.

2.7. Random Forest Regressor (RFR)

Zainab et al. (2020) en un conjunto de B árboles, el valor predicho es el valor medio de las predicciones de todos los árboles individuales. La muestra *bootstrap* elegida a partir de los datos de entrenamiento ha crecido hasta su tamaño máximo utilizando el CART *(Classification and Regression Trees)*. El rendimiento de división del árbol se mejora al considerar típicamente la raíz cuadrada de los descriptores en lugar del conjunto completo. También, se mejora el rendimiento, ya que el bosque aleatorio no realiza ninguna poda para obtener la complejidad adecuada del modelo, la validación cruzada en el bosque aleatorio se realiza en paralelo durante el entrenamiento, validación cruzada, los 2/3 de las muestras se toman para el entrenamiento y 1/3 para la prueba, el modelo no requiere ninguna validación cruzada adicional para probar el rendimiento del modelo.

El modelo RFR es un modelo de aprendizaje automático basado en conjuntos de árboles de decisión para tareas de regresión. La eficiencia de RFR depende de ajustar los parámetros del modelo, dentro de los principales parámetros se puede mencionar:

- Parámetros principales
    - *n_estimators:* Número de árboles en el bosque.
    - *criterion:* Función para medir la calidad de la división en los nodos. En regresión, por defecto es *squared_error* (error cuadrático medio).
    - *max_depth:* Profundidad máxima de los árboles.

- *min_samples_split:* Número mínimo de muestras necesarias para dividir un nodo.
- *min_samples_leaf:* Número mínimo de muestras en una hoja.
- *max_features:* Número máximo de características consideradas en cada división. Valores comunes:
  - *auto o sqrt:* Usa la raíz cuadrada del número total de características.
  - log2: Usa el logaritmo en base 2 del número total de características.
- *bootstrap:* Si es True, usa muestreo con reemplazo para entrenar cada árbol. Si es False, usa todas las muestras disponibles.

- Parámetros de control de crecimiento del árbol
  - *max_leaf_nodes:* Número máximo de hojas en cada árbol.
  - *min_weight_fraction_leaf:* Fracción mínima de peso de la suma total.
  - *max_samples:* Tamaño de la muestra usada para entrenar cada árbol cuando *Bootstrap = True.*
- Parámetros de aleatorización y optimización
  - *random_state:* Controla la aleatoriedad del modelo para obtener resultados reproducibles.
  - *n_jobs:* Número de procesadores utilizados en paralelo.
  - *verbose:* Muestra información del entrenamiento si es > 0.
  - *warm_start:* Si es *True*, reutiliza árboles anteriores y agrega nuevos, en lugar iniciar de cero.

2.8. Multilayer Perceptron Regressor (MLPR)

Müller & Guido (2017) los MLP pueden verse como generalizaciones de los modelos lineales que realizan múltiples etapas de procesamiento para llegar a una decisión, la predicción de una regresión lineal:

Ecuación 1 *Modelo de regresión lineal en su forma general*

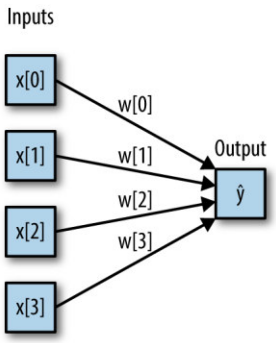
$$\hat{y} = w[0] * x[0] + w[1] * x[1] + \cdots + w[p] * x[p] + b$$

ŷ es una suma ponderada de las características de entrada x[0] a x[p], ponderada por los coeficientes aprendidos w[0] a w[p].

Este modelo tiene muchos más coeficientes (pesos) que aprender, hay uno en cada entrada y en cada unidad oculta, y otro en la capa de salida.

Figura *1*

Visualización de la regresión logística, donde las características de entrada y las predicciones se muestran como nodos, y los coeficientes son las conexiones entre los nodos.



display(mglearn.plots.plot_logistic_regression_graph())

Fuente: (Müller & Guido, 2017, p.105)

2.9. K-Nearest Neighbors Regressor (KNNR)

Morales España et al. (2008) mencionaron que es un método de aproximación simple no paramétrica, basado en la regla del vecino más cercano, que consiste en estimar el valor de un dato desconocido a partir de las características del dato más próximo, según una medida de similitud o distancia, generalmente se usa como algoritmo de clasificación, partiendo de la suposición de que se pueden encontrar puntos similares cerca uno del otro.

En la práctica el algoritmo *KNN Regressor (K-Nearest Neighbors Regressor)* es una variante del algoritmo KNN utilizada para problemas de regresión, en los que el objetivo es predecir un valor numérico en lugar de una categoría o etiqueta. Los principales parámetros KNNR de la biblioteca *scikit-learn* son:

- *n_neighbors:* Número de vecinos a considerar.
- *weights:* Define cómo se ponderan los vecinos en la predicción.
    - *uniform:* Todos los vecinos tienen el mismo peso.
    - *distance:* Los vecinos más cercanos tienen mayor peso.
    - También se puede proporcionar una función personalizada.
- *algorithm:* Método utilizado para encontrar los vecinos más cercanos.
    - *auto:* Selecciona automáticamente el mejor algoritmo según los datos.
    - *ball_tree:* Usa una estructura *Ball Tree* para búsquedas eficientes.
    - *kd_tree:* Usa un *KD-Tree*, eficiente en datos de baja dimensión.
    - *brute:* Usa búsqueda por fuerza bruta (comparando todas las distancias).
- *leaf_size:* Tamaño de la hoja en estructuras *BallTree o KDTree*. Afecta la velocidad de consulta y la memoria.
- p: Define la métrica de distancia de Minkowski:
    - p = 1 Distancia de Manhattan.
    - p = 2 Distancia Euclidiana (por defecto).

  - p > 2 Otras métricas de distancia.
- *metric:* Tipo de métrica de distancia utilizada.
- *n_jobs:* Número de núcleos de CPU usados en el cálculo.

2.10. Autoregressive Integrated Moving Average (Auto-ARIMA)

Es un modelo estadístico que utiliza variaciones y regresiones de datos estadísticos con el fin de encontrar patrones para una predicción, es un modelo dinámico de series temporales, es decir, las estimaciones futuras vienen explicadas por los datos del pasado y no por variables independientes.

Auto-ARIMA *(AutoRegressive Integrated Moving Average)* es un algoritmo que automatiza la selección de los parámetros óptimos de un modelo ARIMA para series temporales. Su objetivo es encontrar los mejores valores para *p, d y q,* así como para los parámetros estacionales *P, D, Q y s* en caso de una serie temporal con estacionalidad (Box et al., 2015).

Parámetros principales de Auto-ARIMA

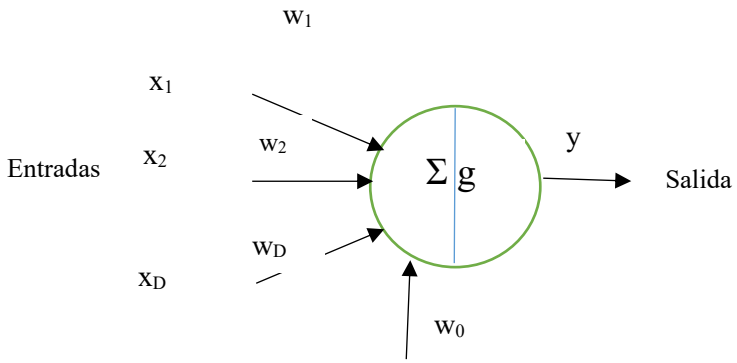
- *p (Auto-Regressive, AR):* Número de rezagos de la variable dependiente en el modelo.
- *d (Integrated, I):* Número de diferenciaciones necesarias para hacer la serie estacionaria.
- *q (Moving Average, MA):* Número de rezagos de los errores en el modelo.
- *P, D, Q (Parámetros estacionales):* Análogos a *p, d y q,* en el contexto de estacionalidad.
- *s:* Periodo estacional de la serie.

2.11. Red de Neuronas Artificiales (RNA)

Palma Méndez & Marín Morales (2008) es un paradigma de procesamiento de información inspirado en el funcionamiento del cerebro. Las RNA están compuestas por un cierto número de elementos de procesamiento o neuronas que trabajan simultáneamente para resolver un problema específico están basadas en el modelo matemático propuesto por McCulloch y Pitts en 1943.

Figura 2

Modelo neuronal de McCulloch-Pitts



Palma Méndez & Marín Morales (2008) el primer paso para obtener la salida *"y"* de la neurona es calcular la suma ponderada *"a"* de las entradas, denominada *activación* de la neurona:

Ecuación 2 *Combinación lineal ponderada –Red Neuronal Artificial (neurona)*

$$a = \sum_{i=1}^{D} w_i x_i + w_0$$

Donde: a es la salida de la combinación lineal, $xí$ representa las variables de entrada, w_i son los pesos asociados a cada entrada x_i, w_0 es un término de sesgo o bias, D es el número total de características de entrada, luego, a partir de este valor *"a"* se obtiene la salida *"y"* de la neurona mediante la aplicación de una función, llamada *función de activación g(a)*, es decir:

Ecuación 3 *Modelo matemático de una neurona en una red neuronal artificial*

$$y = g(a) = g\left(\sum_{i=1}^{D} w_i x_i + w_0\right) = g\left(\sum_{i=0}^{D} w_i x_i\right)$$

es posible tratar el umbral w_0 como un peso más si se supone una entrada añadida x_0 con un valor fijo de 1, g(a) es la función de activación, que introduce no linealidad en el modelo, y es la salida de la neurona después de aplicar la función de activación.

Palma Méndez & Marín Morales (2008) la función de transferencia empleada en este modelo básico de McCulloch-Pitts es la función escalón según las ecuaciones:

Ecuación 4 *Función de transferencia*

$$g(a) = \begin{cases} 0 \; cuando \; a < 0 \\ 1 \; cuando \; a \geq 0 \end{cases}$$

Funciones más comunes:

Ecuación 5 *Función lineal*

$$g(a) = a$$

Ecuación 6 *Función sigmoidea*

$$g(a) = \frac{1}{1 + e^{-a}}$$

Ecuación 7 *Función tangente hiperbólica (tanh)*

$$g(a) = tanh = \frac{e^{a} - e^{-a}}{e^{a} + e^{-a}}$$

Ecuación 8 *Función gaussiana*

$$g(a) = exp\left(-\frac{(a-\mu)^2}{2\sigma^2}\right)$$

2.12. El Perceptrón

La arquitectura que Rosenblatt definió para el perceptrón consistía en una primera capa de *"j"* neuronas con funciones *"φj"* que se encargaban de transformar los datos de entrada. Estas funciones reciben un subconjunto aleatorio de entradas a través de unos pesos fijos y les aplican una función de activación de tipo escalón.

De nuevo existe un peso especial o sesgo w_0 y, de igual modo que en la ecuación se define una entrada artificial x_0 con valor 1 asociado a este peso, en el caso del perceptrón se considera una función de activación extra $\acute{\emptyset}_0 = 1$.

La salida de esta primera capa de funciones $\acute{\emptyset}_j$ se conecta a través de unos pesos a una última neurona para, que finalmente, calcular la salida del perceptrón como:

Ecuación 9 *Salida del perceptrón*

$$y = g(a) = g\left(\sum_{j=0}^{M} w_j \acute{\emptyset}_j(X)\right) = g\left(W^T \acute{\emptyset}\right)$$

donde $\acute{\emptyset}$ es el vector formado por las funciones de activación $\acute{\emptyset}_0, \ldots, \acute{\emptyset}_M$ y *"g"* es la función escalón definida para los valores $\{-1, 1\}$.

2.13. Aprendizaje Supervisado

A. Ramirez Gil & Ramirez Gil (2023) el aprendizaje supervisado se usa típicamente en clasificación y regresión, los más populares son los siguientes: máquinas de vectores de soporte, bayesiano ingenuo, análisis discriminante lineal, árboles de decisión, algoritmo del vecino más cercano, redes neuronales (perceptrón multicapa) y aprendizaje de similitud.

Regresión para ajustar los datos con un modelo de mejor ajuste, los algoritmos más populares son la regresión lineal, la regresión logística, regresión polinomial.

2.14. Aprendizaje No Supervisado

A. Ramirez Gil & Ramirez Gil (2023) en aprendizaje no supervisado, los modelos se alimentan con datos no etiquetados, se usan típicamente para agrupar y asociar, por Agrupación significa dividir los datos en grupos, el algoritmo más popular es el agrupamiento de K-medias y por Asociación significa descubrir reglas que describen la porción mayoritaria de los datos.

2.15. Aprendizaje Semisupervisado

A. Ramirez Gil & Ramirez Gil (2023) en el aprendizaje semisupervisado, se utilizan datos etiquetados como datos no etiquetados, el procedimiento básico es agrupar los datos en diferentes grupos usando un algoritmo de aprendizaje no supervisado y luego usar los datos etiquetados existentes para etiquetar el resto de los datos no etiquetados, los algoritmos más populares incluyen el autoaprendizaje, los métodos generativos, los modelos mixtos y los métodos basados en gráficos.

2.16. Aprendizaje por Refuerzo.

Alonso Ramirez Gil & Ramirez Gil (2023) el aprendizaje por refuerzo, los algoritmos aprenden a encontrar, a través de prueba y error, qué acciones pueden producir la máxima recompensa acumulativa, se usa ampliamente en robótica, videojuegos.

2.17. Metaheurísticos Populares

Descenso por Gradiente (*Hill—Climbing o basic local search*).

Meseguer Gonzalez & Lopez de Mantaras Badia (2017) consiste en moverse a un estado vecino con menor valor de la función objetivo que el estado actual. Esta técnica presenta el problema de convergencia a mínimos locales, buena parte del trabajo en metaheurísticas se ha centrado en como escapar de estos mínimos locales.

2.18. Árboles de Decisión

Benitez Iglesias (2014) un árbol de decisión es una forma de representar reglas de clasificación inherentes a los datos, con una estructura en árbol n-ario que particiona los datos de manera recursiva. Cada rama de un árbol de decisión representa una regla que decide entre una conjunción de valores de un atributo básico (nodos internos) o realiza una predicción de la clase (nodos terminales).

2.19. Poda de los Árboles de Decisión

Benitez Iglesias (2014) el objetivo de la poda de los árboles de decisión es obtener árboles que no tengan en las hojas reglas que afecten al conjunto de entrenamiento.

2.20. Series de Tiempo

De Castro Felippetto (2001) una serie de tiempo se refiere a un conjunto de observaciones ordenadas en el tiempo, éstas no están espaciadas necesariamente, presentan dependencia serial, lo cual implica, dependencia entre instantes de tiempo. La notación usada para denotar una serie temporal es S_1, S_2, S_3 ..., S_T que indica una serie de tamaño T.

García Diaz (2016) Tendencia (T): es una componente de la serie que refleja su evolución a largo plazo, puede ser de naturaleza estacionaria o constante, lineal o exponencial. Componente cíclica (C): recoge las oscilaciones periódicas de amplitud superior a un año, estas

oscilaciones no son regulares, se dan en fenómenos económicos (crecimiento o recesión). Componente estacional (S): recoge oscilaciones que se producen en periodos de repetición iguales o inferiores a un año, inicialmente en series con datos mensuales. Componente aleatorio o irregular (I): es una componente de la serie temporal que recoge las fluctuaciones de eventos imprevisibles.

2.21. Modelos ARIMA

García Diaz (2016) se basa en la estructura de la correlación de los procesos estocásticos estacionarios, para modelizar y predecir series estacionarias o no estacionarias Box y Jenkins en 1970 desarrolló una clase de modelos lineales conocidos como autorregresivos integrados de medias móviles (ARIMA), hay casos especiales de esta metodología son los modelos que relacionan una variable salida con una o más variables entrada, conocidos como modelos de función de transferencia o de regresión dinámica.

Garcia Jimenez (2014) una serie temporal está conformado por un conjunto de N observaciones tomadas de una variable dependiente a lo largo del tiempo, lo cual conforma un proceso que se operacionaliza como una serie de tiempo.

Cáceres Hernández et al. (2007) una serie temporal es un conjunto de observaciones referidas a una magnitud y ordenadas en el tiempo. Pueden ser analizadas con una finalidad descriptiva, si sólo se pretende describir el comportamiento registrado en el pasado, explicativa, si se intenta probar estadísticamente la existencia de relaciones dinámicas causa-efecto entre variables, o predictiva, cuando el objetivo es reducir el grado de incertidumbre sobre el futuro a partir del conocimiento del pasado, para comprender la serie temporal, conviene, identificar

los componentes que dan lugar a los valores observados en el tiempo, y considerar diferentes modelos tanto de cada uno de dichos componentes como del modo en que éstos interactúan.

2.22. Métrica de evaluación de resultados basado en el Error Porcentual Absoluto Medio (MAPE)

La métrica que es utilizada para la medición del desempeño de los modelos predictivos es denominada Métrica del Error Porcentual Absoluto Medio (MAPE), el cual se determina de la siguiente manera:

Ecuación 10 *Métrica del Error Porcentual Absoluto Medio*

$$MAPE = \sum_{t=1}^{n} \frac{|Yr_t - Yexp_t|}{|Yr_t|} * \frac{100\%}{n}$$

Donde:

- Yr_t es el valor real de variable en el periodo t,
- $Yexp_t$ el valor obtenido por el modelo para el periodo t
- n el número total de datos con los que se está trabajando.

Fuente, citado en Kim & Kim (2016)

III. MÉTODO

3.1. Tipo de investigación

- La investigación es de tipo aplicada de acuerdo a la orientación, ya que se enfoca en resolver un problema real aplicando un modelo inteligente predictivo de redes neuronales.
- De acuerdo a la técnica de contrastación, la investigación es explicativa, puesto que se trata de encontrar las causas del problema y su análisis respectivo.
- De acuerdo con el tipo de fuente de recolección de datos, la investigación es retrospectiva ya que la información analizada fue de los archivos publicados por la SUNAT desde el año 2000 a 2023 con los criterios propios y para fines específicos.
- La investigación es longitudinal de acuerdo a la evolución del fenómeno estudiado, puesto que las variables se medirán en varios estadios y comportamientos al que serán expuestos.
- La investigación tiene un enfoque cuantitativo.

3.2. El nivel de investigación

Supo (2023) el nivel de investigación corresponde al predictivo que pueda calcular la probabilidad de ocurrencia de un fenómeno, hecho o acontecimiento.

3.3. Diseño de la investigación

El diseño de la investigación es tecnológica, cuasi experimental, en la que se tomó en cuenta la representatividad de los datos de la muestra que no han sido asignados con un criterio aleatorio y por el desarrollo de algoritmos inteligentes de predicción para la optimización de un proceso de aprovisionamiento económico (De La Cruz Casaño, 2016; Fernández-García et al., 2014)

3.4. Población y muestra

3.4.1. Población

La población está conformada por toda la información reportada por la SUNAT desde el año 2000 a 2023, respecto de los ingresos recaudados en millones de soles anual/mensual[1].

3.4.2. Tamaño de la Muestra

El método a usar para el muestreo en la investigación es no probabilístico, de tipo deliberado, los elementos que integran la muestra son todos los elementos de la población correspondiente a los años 2000 a 2023.

Oddi et al. (2018) respecto de error de medición que pudiera tener la data, se puede evidenciar que la información de recaudación tributaria es público ubicados en la página web de SUNAT, por lo que el Sesgo de Selección de datos no aplicaría en este caso. Respecto del Sesgo Temporal, se puede evidenciar que los datos provienen de la data de los años 2000 a 2023 años que SUNAT ha publicado la recaudación correspondiente de la época de crisis de salud pública COVID 19, por tanto, el sesgo temporal no aplicaría. Los modelos de aprendizaje supervisado desarrollados en la tesis poseen técnicas de validación cruzada y ajuste de hiperparámetros.

3.5. Operacionalización de variables

Variable de dependiente

- Predicción de la recaudación tributaria

[1] https://www.sunat.gob.pe/estadisticasestudios/ingresos-recaudados.html

Variable independiente

- Algoritmo Estructura Multilayer Perceptrón Optimizado - *Hill Climbing (MLP-O)*
- *Multi-Layer Perceptron Regressor* (MLPR)
- *Gradient Boosting Regressor* (GBR)
- *Random Forest Regressor* (RFR)
- *K-Nearest Neighbors Regressor* (KNNR)
- *Autoregressive Integrated Moving Average* (AutoARIMA)

3.6. Procedimientos

3.6.1. ***Identificación de la Estructura MLP Óptima***.

La integración entre las redes *multilayer perceptron* y el algoritmo de *Hill Climbing*.

a. Se genera de forma aleatoria el individuo (posible solución), en seguida, se evalúa su desempeño por una red neuronal *multilayer perceptrón*. Es decir, se obtiene el desempeño utilizando una métrica (MAPE).
b. Se genera un número de forma aleatoria de la misma longitud del individuo para posicionarse y escoger a uno o dos vecinos del individuo y generar el cambio de valor.
c. En caso se posicione en un valor extremo, solo se obtendría un nuevo individuo, y en caso de ser un valor central, se crearán dos nuevos individuos.
d. En caso de crearse individuos válidos, y que no existan en la matriz de almacenamiento. Entonces se evalúan y se obtienen sus MAPEs correspondientes.
e. Cuando se encuentre un individuo con un MAPE menor al del actual individuo entonces ese se convertirá en el individuo actual, caso contrario se genera aleatoriamente otro individuo.

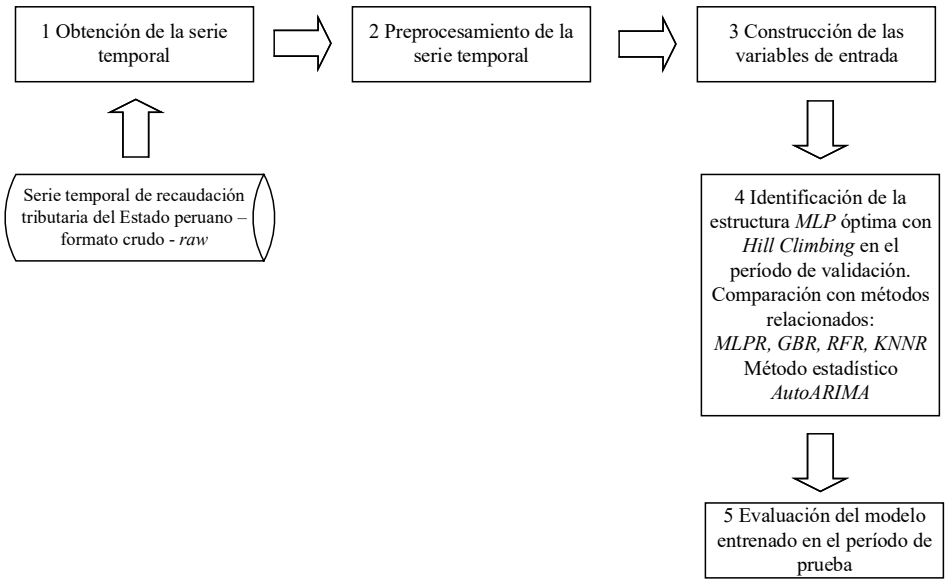
*3.6.2. **Modelos de Regresión para Series Temporales, Hyperparameter Tuning**.*

a. Transforma los datos en formato de series temporales.
b. Configura los modelos de regresión clásica como modelos de regresión para series temporales.
c. Para cada modelo de regresión identifica y configura los parámetros con sus respectivos intervalos de valores.
d. Para cada modelo configurado ejecuta la técnica de *Hyperparameter Tuning* utilizando el subconjunto *Train* y con validación cruzada con tamaño de ventana 12.
e. Obtiene la mejor combinación de parámetros.
f. Entrena los modelos con los valores de los parámetros identificados utilizando el subconjunto *Train*.

3.6.3. Desarrollo de la Metodología

En la Figura 3, se muestra la metodología propuesta, el cual contiene cinco (5) fases: obtención de la serie temporal, preprocesamiento de la serie temporal, construcción de las variables de entrada, identificación de la estructura MLP óptima con el algoritmo *Hill Climbing* en el período de validación, comparación con métodos relacionados *MLPR, GBR, RFR, KNNR,* método AutoARIMA y predicción en el período de prueba o *recall.*

Figura 3 *Metodología propuesta*



3.7. Análisis de datos

3.7.1. Obtención de la Serie Temporal

La serie temporal fue extraída del *website* de la SUNAT[2], la serie utilizada corresponde a la suma de los impuestos I, II, III, IV:

I. Ingresos Tributarios recaudados por la SUNAT - Internos

II. Ingresos Tributarios recaudados por la SUNAT - Aduaneros

III. Contribuciones Sociales

IV. Ingresos No Tributarios

El resultado que se espera obtener a partir de los dados en formato excel crudo (raw) es ilustrado en la Tabla 1.

[2] : https://www.sunat.gob.pe/estadisticasestudios/ingresos-recaudados.html

Tabla 1

Ilustración del proceso de transformación de conjunto de datos para el formato de series temporales, donde xxxx xxx xxx.xx representa números en millones de soles.

Raw format				**Times series format**	
				Id	Income
	Enero	…	Diciembre	2000/01	xxxx xxx xxx.xx
2000	xxxx xxx xxx.xx	xxxx xxx xxx.xx	xxxx xxx xxx.xx	2000/02	xxxx xxx xxx.xx
2001	xxxx xxx xxx.xx	xxxx xxx xxx.xx	xxxx xxx xxx.xx	…	…
…	…	…	…	2023/11	xxxx xxx xxx.xx
2022	xxxx xxx xxx.xx	xxxx xxx xxx.xx	xxxx xxx xxx.xx	2023/12	xxxx xxx xxx.xx
2023	xxxx xxx xxx.xx	xxxx xxx xxx.xx	xxxx xxx xxx.xx		

3.7.2. *Preprocesamiento* de la Serie Temporal

Los valores obtenidos de la serie temporal contienen valores altos, de esa forma es necesario normalizar la serie temporal por normalización Min-Max. Para la normalización se usó la Ecuación 11.

Ecuación 11 *Ecuación de normalización min-max*

$$Y_{jn} = \frac{y_{j-}Y_{min}}{y_{max-}Y_{min}}$$

Donde:

- Y_{jn} : valor normalizado de la observación j
- Y_j : valor actual para la observación j
- Y_{min} : valor mínimo de la variable
- Y_{max}: valor máximo de la variable

Fuente: adaptado de Faceli et al., (2011)

Interpretación del resultado:

Si $y_j = Y_{min}$, entonces $Y_{jn} = 0$.

Si $y_j = Y_{max}$, entonces $Y_{jn} = 1$.

Los valores intermedios estarán en el rango [-1,1].

3.7.3. Construcción de las Variables de Entrada

Mamani Ticona et al. (2017) para la construcción de las variables de entrada, se dividió las variables de entrada de la red en tres grupos: (a) media móvil, (b) diferencias móviles y (c) valores pasados (atrasos) de la serie:

1. La media móvil mide el valor medio en un determinado periodo (Mendelsohn, 2000), fueron utilizados dos entradas MM2 (últimos dos meses) y MM3 (últimos tres meses).
2. Diferencias para identificar las tendencias, fueron utilizados dos diferencias D(1-2) y D(1-3), donde D(1-2) significa la diferencia entre el último y el penúltimo valor conocido, y D(1-3) significa la diferencia entre el último y el antepenúltimo valor conocido.
3. Fueron seleccionados seis entradas con valores pasados: M-1, M-2, M-3, M-6, M-12 y M-24, donde M-1, M-2, M-3 indica el ultimo, penúltimo y el antepenúltimo valor conocido, anterior a los valores que predecirá, M-6 indica el valor seis meses anteriores a los valores que se predecirán, M-12 son los valores anteriores a los meses correspondientes para la predicción de un año.

3.7.4. Identificación de la Estructura MLP Óptima utilizando el Algoritmo Hill Climbing (MLP-O)

Se propone la integración entre la Red Multilayer Perceptrón y el algoritmo de *Hill Climbing*. En seguida, se explica la integración.

1. Se genera de forma aleatoria un individuo (posible solución), en seguida, se evalúa su desempeño por una red neuronal *multilayer perceptrón*. Es decir, se obtiene su desempeño utilizando una métrica (MAPE). Luego, tanto el individuo como su GMAPE son almacenados en una matriz.
2. Se genera un número de forma aleatoria de la misma longitud del individuo para posicionarse y escoger a uno o dos vecinos del individuo y genera el cambio de valor.
3. En caso se posicione en un valor extremo, solo se obtendría un nuevo individuo, y en caso de ser un valor central, se crearán dos nuevos individuos.
4. En caso de crearse individuos válidos, y que no existan en la matriz de almacenamiento. Entonces se evalúan y se obtienen sus MAPEs correspondientes.
5. Cuando se encuentre un individuo con un MAPE menor al del actual individuo entonces ese se convertirá en el individuo actual, caso contrario se genera aleatoriamente otro individuo.

3.7.5. *Predicción en el Periodo de Prueba*

Como estudio de caso, fue realizada la predicción *multi-step* 12 meses para delante de los ingresos recaudados por la SUNAT del año 2023. En esta fase se identifican los mejores pesos y *bias*. Para identificar los mejores pesos y *bias*, las estructuras de redes MLP identificadas en la sección anterior, son ejecutadas con 1000 experimentos.

3.7.6. Comparación con otros modelos de aprendizaje supervisado

Esta etapa corresponde a la ejecución de tareas de comparación con dos modelos identificados en la revisión bibliográfica y que servirán para realizar comparaciones con el método desarrollado, MLP-O.

3.7.7. ***Métodos de optimización de parámetros basado en búsqueda aleatoria***

Para ejecutar el experimento de comparación con otros métodos, la técnica de *Hyperparameter Tuning* fue utilizada como base para configurar algunos modelos de regresión para series temporales. El ajuste de hiperparámetros, es una técnica fundamental en el área de inteligencia artificial que tiene como objetivo optimizar el desempeño de un determinado modelo de aprendizaje máquina. Los hiperparámetros son parámetros de los modelos que no son aprendidos directamente en el proceso de entrenamiento, pero sí son definidos manualmente antes que inicie el entrenamiento. Es importante obtener los mejores hiperparámetros porque diferentes combinaciones de parámetros pueden traer resultados significativos de desempeño de los modelos. Por lo tanto, las técnicas de *Hyperparameter Tuning* tienen como objetivo maximizar la precisión y al mismo tiempo minimizar el error, evitando a su vez el *overfitting* y *underfitting* (Sobreajuste y Subajuste).

Para realizar la comparación, los mejores parámetros de algunos modelos clásicos de regresión configurados ahora para regresión de series temporales e implementados en la librería *scikit-learn*, fueron obtenidos utilizando la técnica de *hyperparameter tuning* basada en la búsqueda aleatoria (*ForecastingRandomizedSearchCV)* de la librería *sktime* SKTIME (2024), que es una extensión de *Scikit-learn* especializada en series temporales y pronósticos *(forecasting)*, de esta forma, los modelos que incluyen la técnica *Hyperparameter Tuning* en su proceso de entrenamiento completan los pasos del Algoritmo A1.

Tabla 2

Algoritmo A1

Algoritmo para entrenar modelos de regresión para series temporales utilizando *Hyperparameter Tuning* de búsqueda aleatoria.
1. Transformar los datos en formato de series temporales

2. Configurar los modelos de regresión clásica como modelos de regresión para series temporales.
3. Para cada modelo de regresión identificar y configurar los parámetros con sus respectivos intervalos de valores.
4. Para cada modelo configurado ejecutar la técnica de *Hyperparameter Tuning* utilizando el subconjunto *Train* y con validación cruzada con tamaño de ventana 12.
5. Obtener la mejor combinación de parámetros.
6. Entrenar los modelos con los valores de los parámetros identificados utilizando el subconjunto *Train*.

3.7.8. Método automático/estadístico Auto-ARIMA

Autoregressive Integrated Moving Average (Auto-Arima) SKTIME (2021) este método estadístico fue considerado en el experimento por ser semejante con nuestro método en la forma de calcular la mejor configuración de sus parámetros. Este método no utiliza el cálculo de parámetros por *hyperparameter tuning,* puesto que, posee algoritmos con base en estadística que permiten calcular automáticamente la mejor combinación de parámetros.

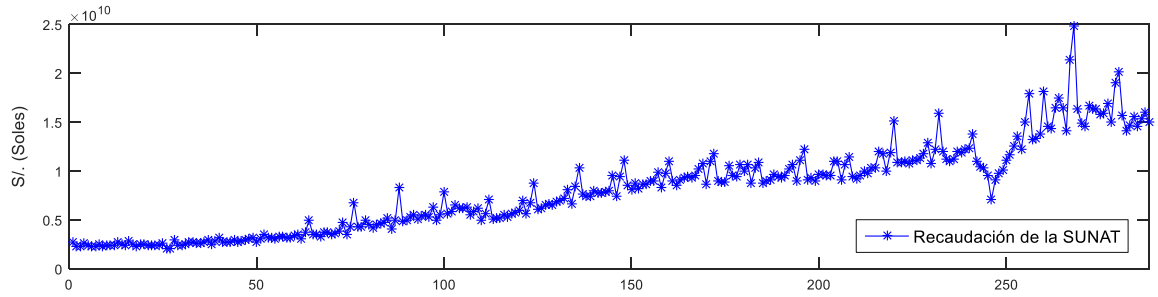
IV. RESULTADOS

4.1. Obtención de la serie temporal

La trayectoria de la serie temporal es mostrada en la Figura 4, como se puede observar, el periodo de la serie va del mes de enero de 2000 a diciembre de 2023.

Figura 4

Trayectoria de la serie de los ingresos recaudados por la SUNAT 2000-2023



Los periodos de la serie temporal fueron divididos en tres bloques para distintas finalidades:

- Entrenamiento (enero de 2000 a diciembre del 2022)
- Validación (enero de 2000 a diciembre de 2022)
- Prueba o *test* (enero a diciembre de 2023)

4.2. Preprocesamiento de la serie temporal

Para el preprocesamiento de la serie se utilizó la Ecuación 11. En esta etapa se normalizaron los valores de la serie temporal en un rango de -1 a 1.

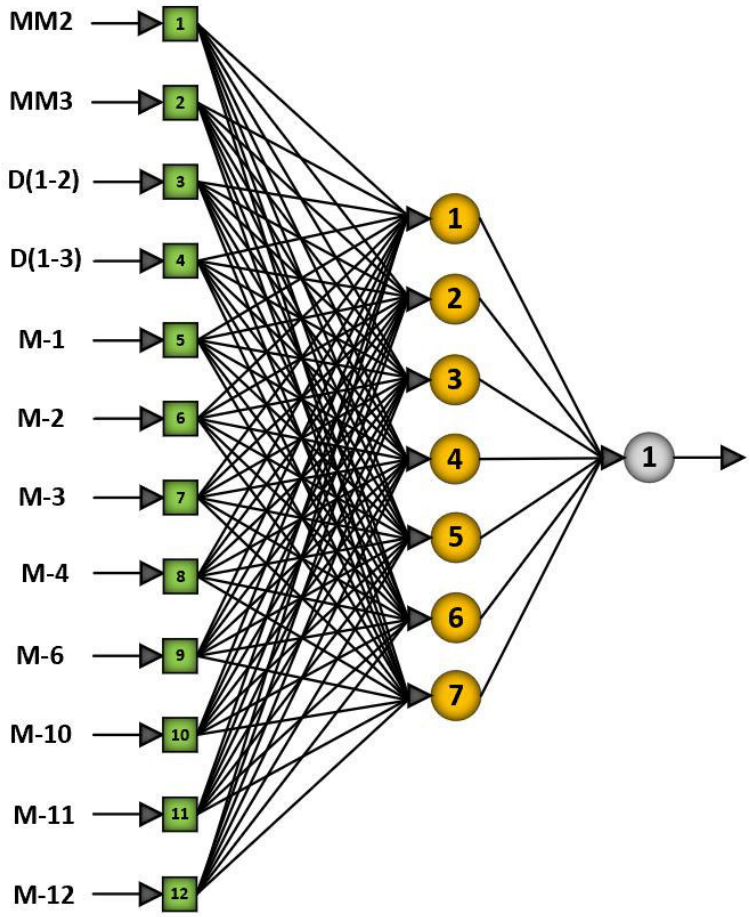
4.3. Construcción de las variables de entrada

Para realizar la predicción fue aplicado el algoritmo propuesto, en cada experimento las redes son entrenadas y validadas. Las variables de entrada de las redes fueron construidas con

las ecuaciones de la Tabla 3, la estructura de red neuronal con 12 entradas, 7 neuronas en la capa oculta y una neurona de salida, es mostrada en la Figura 5.

Figura 5

Variables de entrada de la red MLP



En la Tabla 3, es mostrada las variables de entrada con sus respectivas ecuaciones para cada variable de entrada.

Tabla 3

Variables de entrada de las redes

Variables de entrada	Procesamiento de los valores de la serie (S)
$MM2$	$\frac{S_{t-1} + S_{t-2}}{2}$
$MM3$	$\frac{S_{t-1} + S_{t-2} + S_{t-3}}{3}$
$D(1-2)$	$S_t - S_{t-1}$
$D(1-3)$	$S_t - S_{t-2}$
$M-1$	S_{t-1}
$M-2$	S_{t-2}
$M-3$	S_{t-3}
$M-4$	S_{t-4}
$M-6$	S_{t-6}
$M-10$	S_{t-10}
$M-11$	S_{t-11}
$M-12$	S_{t-12}

4.4. Identificación de la Estructura MLP Óptima en el Período de Validación

El objetivo en esta fase es encontrar la mejor estructura de red MLP en el periodo de validación (enero de 2000 a diciembre de 2022). Para encontrar la mejor estructura MLP (identificación de entradas y número de neuronas en la capa oculta) se utilizó la estructura combinada basada en redes *multilayer perceptrón* y el algoritmo de *Hill Climbing.*

4.4.1. Parámetros de la Red Neuronal MLP

Las redes neuronales fueron ejecutadas con los siguientes parámetros:

- Número máximo experimentos: 1000
- Número máximo de fallas en el conjunto de validación: 25
- Como función de activación en la capa oculta fue utilizada la función *tansing (Hyperbolic Tangent Sigmoid)*
- Como función de activación en la capa de salida fue utilizada la función *purelin (Linear)*
- Para el entrenamiento de las redes fue utilizado el método *Levenberg-Marquardt (LM)*
- Número máximo de épocas: 25

4.4.2. Identificación de la Estructura MLP óptima

Para identificar la estructura solución, se realizó la predicción *multi-step* de 12 pasos para adelante de una red neuronal, una capa oculta y número de neuronas variando de 1 a 7. Los resultados de la estructura solución, son mostrados en la Tabla 4. La estructura de la solución posee 5 entradas (MM3, M-1, M-3, M-4, M-12 y 5 neuronas en la capa oculta con 3.383 % de error *MAPE.* (Bastos & Campos, 2010)

Figura 6

Estructura óptima MLP

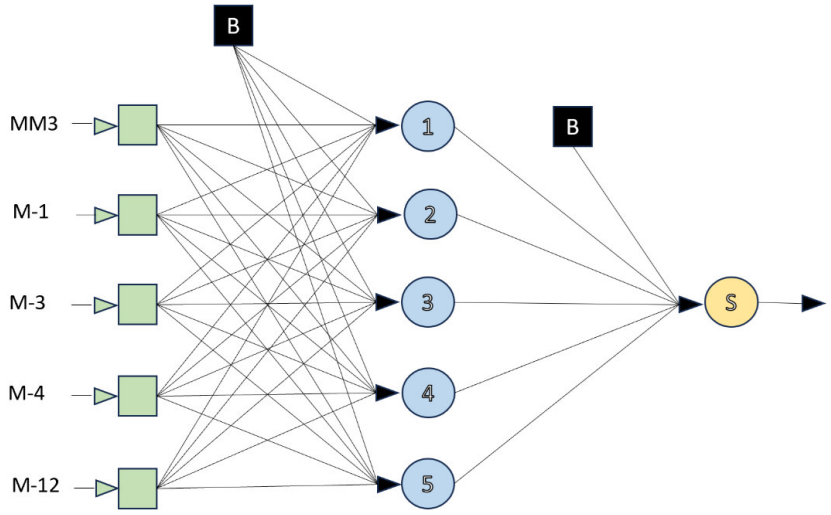


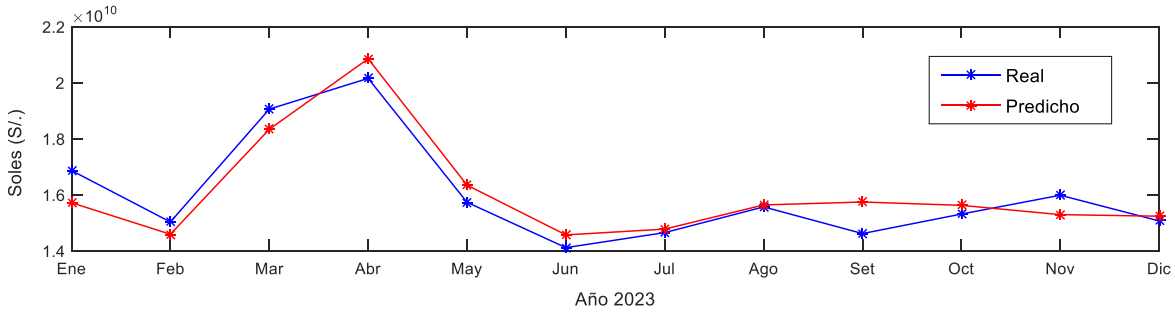
Tabla 4

Resultados de la estructura solución

Variables de entrada de la red neuronal	Número de neuronas	MAPE de la estructura solución
MM3, M-1, M-3, M-4, M-12	5	3.383 %

Figura 7

Trayectorias de los valores reales y predichos de ingresos recaudados 2023



En la Tabla 5, es mostrado el valor real de la recaudación de la SUNAT de enero a diciembre del año 2023 y el valor predicho obtenido por el algoritmo propuesto.

Tabla 5

Comparación de los valores reales y predichos de la recaudación de la SUNAT del año 2023

Meses	Recaudación real de la SUNAT	Predicción *multi-step* del algoritmo propuesto
Enero	16864577392.81	15713808589.71
Febrero	15027128137.53	14589818911.03
Marzo	19054534827.42	18335023994.66
Abril	20153407083.73	20853554728.77
Mayo	15730199663.28	16342988404.86
Junio	14111888890.55	14566081810.90
Julio	14648739285.32	14774637559.09
Agosto	15565972895.45	15635955644.75
Setiembre	14614241126.18	15740290502.10
Octubre	15310267710.61	15620916748.66
Noviembre	15982371790.94	15289594200.23
Diciembre	15065553104.80	15233567521.85
Total	192128881908.62	192696238616.60

4.5. Comparación con otros métodos

Esta sección describe el esquema definido para ejecutar los experimentos de comparación de nuestro método (MLP-O) con otros métodos semejantes que tienen como objetivo obtener los mejores parámetros.

Configuración del conjunto de datos

Para ejecutar los experimentos de comparación, el conjunto de datos de ingresos de la SUNAT fue transformado en formato de series temporales, anexo 1. Luego fue dividido en subconjuntos de la siguiente forma: *Train* (años 2000 hasta el 2022) y *Test* (año 2023). El subconjunto de validación es obtenido directamente del subconjunto *Train* utilizando validación cruzada con tamaño de ventana 12.

Hyperparameter Tuning por búsqueda aleatoria

Para ejecutar el experimento de comparación con otros métodos, la técnica de *hyperparameter tuning* descrita en la sección 2.3 fue utilizada como base para configurar algunos modelos de regresión orientados a series temporales con el objetivo de obtener la mejor configuración de parámetros.

Siguiendo el algoritmo A1 (definido en la metodología, sección 3.5.3), a continuación, se describe el proceso de entrenamiento de los modelos clásicos de regresión A1.

4.5.1. Proceso de entrenamiento

En esta sección se describe la configuración de los modelos de regresión para ejecutar el respectivo entrenamiento con el subconjunto *Train*. La Tabla 6 muestra el tipo de metodología utilizada para los siguientes modelos:

- ***Multi-Layer Perceptron Regressor (MLPR).*** Hinton (1989) este modelo de regresión tiene como base la estructura de red neuronal artificial multicapa (MLP). En la sección 2.8 se puede consultar mayor detalle sobre el funcionamiento de este modelo. Sus parámetros fueron configurados siguiendo el siguiente espacio de valores:
    - Número de neuronas en la capa oculta *(hidden_layer_sizes):* [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21]
    - Método de activación (*activation*): [*'identity'*, *'logistic'*, *'tanh'*, *'relu'*].

      identity, activación sin operación, útil para implementar cuellos de botella lineales.

      logistic, basada en una función sigmoidea logística.

      tanh, basado en una función hiperbólica.

      relu, basado en una función unitaria lineal rectificada

- Optimizador de pesos (*solver*): *['sgd', 'adam']. 'sgd',* se refiere al optimizador basado en el descenso de gradiente estocástico. *'adam'*, se refiere al optimizador de descenso de gradiente estocástico propuesto por Kingma, Diederik, and Jimmy Ba (Kingma & Lei Ba, 2015).
- Parámetro alfa *(alpha):* [0.0001, 0.001, 0.01, 0.05].
- Tasa de aprendizaje *(learning_rate): ['constant', 'invscaling', 'adaptive']. 'constant'* es una constante definida utilizando la tasa de aprendizaje inicial. *'Invscaling'* se refiere a la disminución gradual de la tasa de aprendizaje; *'adaptive'* mantiene la tasa de aprendizaje constante igual a la tasa de aprendizaje inicial siempre que la pérdida de entrenamiento siga disminuyendo.

- ***Gradient Boosting Regressor (GBR).*** Friedman (2001) este modelo se refiere a una clase de algoritmos de aprendizaje máquina que actúan en conjunto (*ensembles*) y que se puede utilizar para problemas de clasificación o regresión. Los *ensembles* se construyen a partir de modelos de árboles de decisión y se ajustan para corregir los errores de predicción cometidos por modelos anteriores. El espacio de valores de parámetros es configurado de la siguiente forma:
  - Nivel máximo de profundidad *(max_depth):* [3, 5, 6, 10, 15, 20]
  - Tasa de aprendizaje *(learning_rate):* [0.01, 0.1, 0.2, 0.3]
  - Porcentaje de submuestra *(subsample):* [0.5, 0.6, 0.7, 0.8, 0.9]
  - Proporción de submuestra de columnas al construir cada árbol *(colsample_bytree):* [0.4, 0.5, 0.6, 0.7, 0.8, 0.9]
  - Proporción de submuestra de columnas para cada nivel *(colsample_bylevel):* [0.4, 0.5, 0.6, 0.7, 0.8, 0.9]

- Número de estimadores (n_estimators): [100, 200, 500, 1000]

- ***Random Forest Regressor (RFR).*** Mansour & Schain (2001) este modelo es categorizado como un *ensemble* de regresión y está basado en conjunto de árboles de decisión donde el espacio de características es dividido en estructura de árboles, donde el valor de predicción es calculado en los últimos nodos hoja del árbol. El espacio de valores de sus parámetros es configurado de la siguiente forma:
  - Número de estimadores *(n_estimators):* [50, 100, 200, 300, 500]
  - Mínimo de muestras por hojas *(min_samples_leaf):* [1, 2, 3, 4, 5]
  - Número máximo de características *(max_features):* [1, 2, 3, 4]
  - Máximo nivel de profundidad del árbol *(max_depth):* [50, 80, 90, 100]

- ***K-Nearest Neighbors Regressor (KNNR).*** Qi et al. (2022), este modelo de regresión está basado en el cálculo de los *K* vecinos más cercanos. En la sección 2.9 se puede consultar mayor detalle sobre el funcionamiento de este modelo. El espacio de valores de sus parámetros fue configurado de la siguiente forma:
  - Número de vecinos más próximos *(n_neighbors):* [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Nuestro método propuesto ***(MLP-O)*** fue configurado y entrenado siguiendo los pasos descritos en la Sección 3.6.3, por lo tanto, es necesario indicar que el método MLP-O no utiliza *Hyperparameter Tuning*, puesto que utiliza una optimización basada en el algoritmo *Hill Climbing.*

Por otro lado, el método *Autoregressive Integrated Moving Average (Auto-ARIMA).* SKTIME (2021b), fue considerado en el experimento por ser semejante con nuestro

método al tener como objeto de cálculo la mejor configuración de sus parámetros. Este método no utiliza el cálculo de parámetros por *Hyperparameter Tuning*, puesto que, posee algoritmos con base en estadística que permiten calcular automáticamente la mejor combinación de parámetros. En la Sección 2.10 se puede consultar mayor detalle sobre el funcionamiento de este modelo.

Un resumen de la configuración de los métodos para realizar el experimento de comparación es ilustrado en la Tabla 6.

Tabla 6

Descripción de modelos utilizados en la ejecución del experimento de comparación

Modelo	Tipo de modelo	Técnica para obtener los mejores parámetros
MLP-O	Aprendizaje máquina	Optimización basada en *Hill Climbing*
MLPR	Aprendizaje máquina	*Hyperparameter Tuning*
GBR	Aprendizaje máquina	*Hyperparameter tuning*
RFR	Aprendizaje máquina	*Hyperparameter tuning*
KNNR	Aprendizaje máquina	*Hyperparameter tuning*
Auto-ARIMA	Estadístico	Implemente algoritmo automático

Una vez encontrados la mejor combinación de parámetros para cada modelo, Tabla 7 muestra los mejores parámetros y modelos que fueron entrenados.

Tabla 7

Resultados de los mejores valores de parámetros para cada modelo obtenidos con la técnica de Hyperparameter Tuning

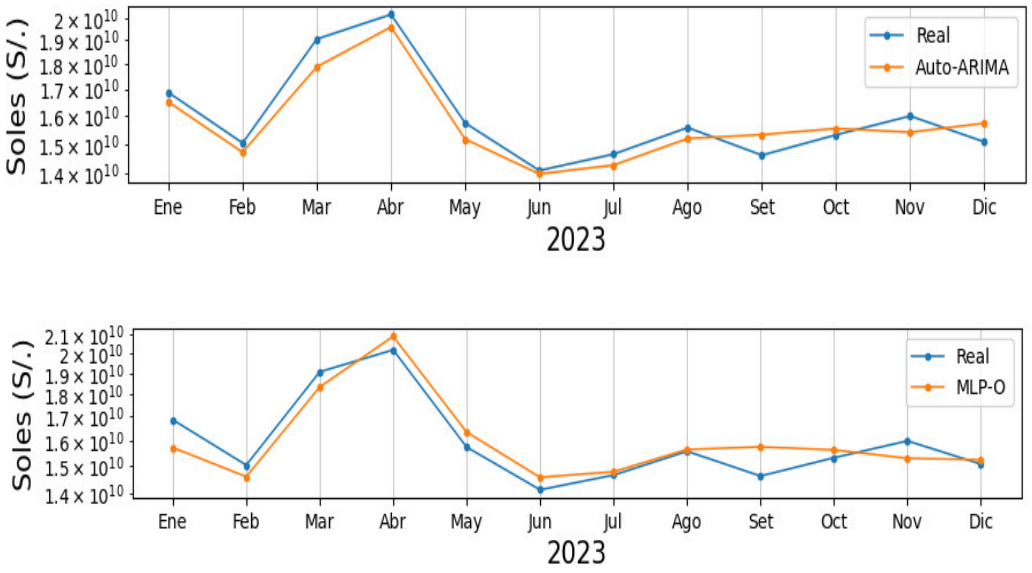
Modelo	Mejores parámetros
MLPR	*'solver': 'adam', 'max_iter': 300, 'learning_rate': 'constant', 'hidden_layer_sizes': (8), 'alpha': 0.0001, 'activation': 'relu'*
GBR	*'subsample': 0.6, 'n_estimators': 200, 'max_depth': 15, 'learning_rate': 0.2, 'colsample_bytree': 0.799, 'colsample_bylevel': 0.899*
RFR	*'n_estimators': 50, 'min_samples_leaf': 1, 'max_features': 4, 'max_depth': 90*
KNNR	*'n_neighbors': 1*

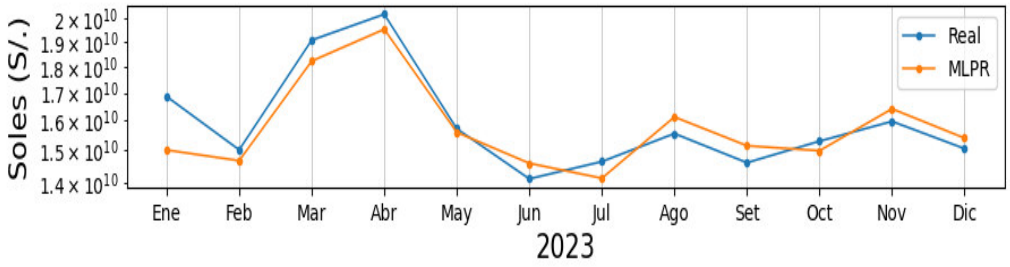
4.5.2. Proceso de Evaluación

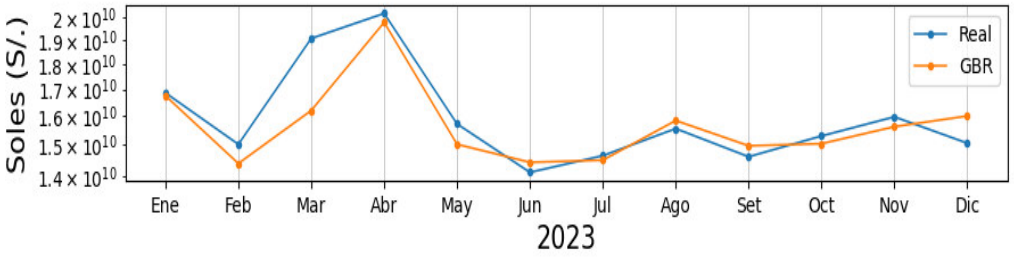
Después del entrenamiento de los modelos, fueron calculados los resultados de predicción sobre el subconjunto Test (datos del año 2023). Los resultados de predicción son visualizados en la Figura 8 y la Tabla 8. Los resultados cuantitativos de comparación utilizando la métrica MAPE son mostrados en la Tabla 9.

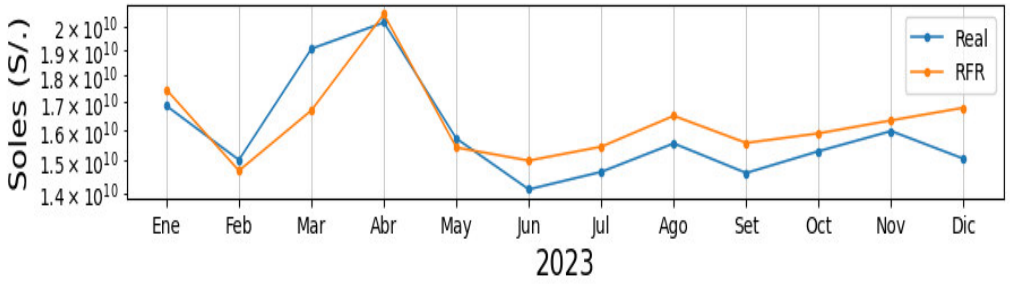
Figura 8

Resultados de predicción para el año 2023. Son visualizados los resultados para comparar los valores reales y los valores predichos por los modelos de regresión




Soles (S/.)
2 x 10^10
1.9 x 10^10
1.8 x 10^10
1.7 x 10^10
1.6 x 10^10
1.5 x 10^10
1.4 x 10^10
Ene Feb Mar Abr May Jun Jul Ago Set Oct Nov Dic
2023
Real
MLPR


Soles (S/.)
2 x 10^10
1.9 x 10^10
1.8 x 10^10
1.7 x 10^10
1.6 x 10^10
1.5 x 10^10
1.4 x 10^10
Ene Feb Mar Abr May Jun Jul Ago Set Oct Nov Dic
2023
Real
GBR


Soles (S/.)
2 x 10^10
1.9 x 10^10
1.8 x 10^10
1.7 x 10^10
1.6 x 10^10
1.5 x 10^10
1.4 x 10^10
Ene Feb Mar Abr May Jun Jul Ago Set Oct Nov Dic
2023
Real
RFR

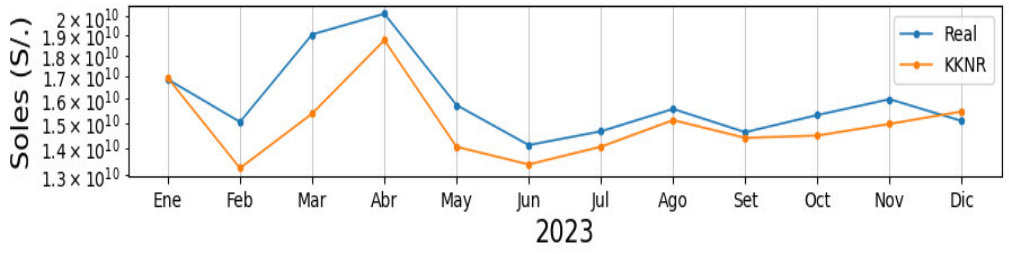

Soles (S/.)
2 x 10^10
1.9 x 10^10
1.8 x 10^10
1.7 x 10^10
1.6 x 10^10
1.5 x 10^10
1.4 x 10^10
1.3 x 10^10
Ene Feb Mar Abr May Jun Jul Ago Set Oct Nov Dic
2023
Real
KKNR

Tabla 8

Resultados de predicción en millones de soles para el año 2023

	Real	Auto-ARIMA	**MLP-O**	MLPR	GBR	RFR	KNNR
Enero	**16864.58**	16501.52	**15713.81**	15020.42	16753.76	17457.51	16929.92
Febrero	**15027.13**	14712.34	**14589.82**	14678.50	14394.85	14685.43	13256.78
Marzo	**19054.53**	17882.41	**18335.02**	18212.20	16211.46	16704.25	15370.54
Abril	**20153.41**	19585.11	**20853.55**	19508.42	19753.68	20514.54	18756.86
Mayo	**15730.20**	15160.30	**16342.99**	15608.24	15028.02	15429.93	14045.62
Junio	**14111.89**	14000.94	**14566.08**	14595.38	14432.32	15005.41	13391.08
Julio	**14648.74**	14287.39	**14774.64**	14130.73	14502.40	15457.90	14048.78
Agosto	**15565.97**	15180.37	**15635.96**	16135.56	15849.21	16510.34	15100.88
Setiembre	**14614.24**	15315.52	**15740.29**	15160.92	14978.18	15582.12	14391.57
Octubre	**15310.27**	15534.90	**15620.92**	14998.08	15047.47	15903.90	14480.91
Noviembre	**15982.37**	15400.57	**15289.59**	16420.98	15633.50	16351.34	14946.53
Diciembre	**15065.55**	15721.15	**15233.57**	15413.59	16011.79	16799.62	15458.04

Tabla 9

Resultados cuantitativos de comparación utilizando la métrica MAPE, año 2023

Modelo	MAPE
AutoARIMA	3.0691 %
MLP-O	**3.3830 %**
MLPR	3.5923 %
GBR	3.6660 %
RFR	5.3389 %
KNNR	6.4466 %

4.6. Contrastación de hipótesis

Para el análisis de normalidad de datos, se aplicó la prueba de *Shapiro Wilk* debido a que el tamaño de la muestra es menor a 5000, para esto se planteó la hipótesis nula y alterna.

4.6.1. ***Hipótesis para probar la Normalidad***

H0: Los datos para los algoritmos provienen de una distribución normal

H1: Los datos para los algoritmos no provienen de una distribución normal

Tabla 10

Pruebas de normalidad

Pruebas de normalidad						
	Kolmogorov-Smirnov[a]			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
AutoArima	0.263	12	0.021	0.848	12	0.035
MLP_O	0.320	12	0.001	0.741	12	0.002
MLRP	0.221	12	0.109	0.840	12	0.028
GRB	0.202	12	0.189	0.791	12	0.007
RFR	0.222	12	0.104	0.825	12	0.018
KNN	0.220	12	0.111	0.880	12	0.088

a. Corrección de significación de Lilliefors

Como podemos observar solo la variable KNN con p-value = 0.088 sus datos provienen de una distribución normal. Los demás datos de las otras variables NO provienen de una distribución normal.

Entonces tenemos que aplicar una prueba estadística no paramétrica para realizar la comparación de los seis algoritmos, la prueba aplicada es de *Kruskal-Wallis*

4.6.2. *Hipótesis general*

- H_o: Si no es posible implementar el modelo algorítmico de inteligencia artificial *MLP– Hill Climbing* para la predicción de la recaudación tributaria del Estado peruano entonces no podemos afirmar que la predicción es mejor respecto de otros modelos de inteligencia artificial.

H_a: Si es posible implementar el modelo algorítmico de inteligencia artificial *MLP– Hill Climbing* para la predicción de la recaudación tributaria del Estado peruano entonces podemos afirmar que la predicción es mejor respecto de otros modelos de inteligencia artificial.

Conclusión:

Según la Tabla 4 se muestra las variables de entrada de la red neuronal (MM3, M-1, M-3, M-4, M-12), 5 neuronas y un MAPE de 3.383 %, en la Tabla 5 se muestra la comparación la predicción de las recaudaciones de la SUNAT para el año 2023 con ello se puede afirmar que fue posible implementar el modelo algorítmico de inteligencia artificial *MLP– Hill Climbing* para la predicción de la recaudación tributaria del Estado peruano. En la Tabla 8 y Tabla 9, se puede apreciar los resultados de predicciones para el año 2023 de los seis modelos, en la que en el grupo de modelos relacionados a inteligencia artificial predictiva el modelo MLP-O (Multilayer perceptrón Optimizado con el algoritmo *Hill Climbing*) es el mejor con un MAPE de 3.3830 % respecto de los otros modelos de IA. Con excepción del modelo estadístico AutoARIMA el resultado fue de 3.0691 % habiendo una diferencia de 0.3139 %. En conclusión, se puede afirmar que se acepta la hipótesis alterna.

4.6.3. Hipótesis específicas

- H_o: No es posible obtener la serie temporal de datos de la recaudación tributaria de la SUNAT entonces no se procesa ni se construye las variables de entrada.

  H_a: Si es posible obtener la serie temporal de datos de la recaudación tributaria de la SUNAT entonces se procesa y construye las variables de entrada.

Conclusión:

Respecto de la obtención de la serie temporal se puede visualizar en la Figura 4 la trayectoria de los ingresos recaudados por la SUNAT 2000 a 2023 divididos en tres bloques: entrenamiento, validación y prueba, respecto del procesamiento se aplicó la normalización de valores de la serie en un rango de -1 a 1, logrando construir las 12 variables de entrada de las redes (Tabla 3), ejecutándose 1000 experimentos, 25 número de fallas en el conjunto de validación, en la capa oculta se utilizó la función *tansing (Hyperbolic Tangent Sigmoid),* 25 épocas. Por tanto, se concluye que se acepta la hipótesis alterna.

- Ho: No es viable elaborar las estructuras de procesamiento de los algoritmos MLP óptimo con *Hill Climbing*, *MLPR, GBR, RFR, KNNR*, AutoARIMA entonces no es factible comparar sus resultados para el periodo de validación.

  Ha: Si es viable elaborar las estructuras de procesamiento de los algoritmos MLP óptimo con *Hill Climbing*, *MLPR, GBR, RFR, KNNR*, AutoARIMA entonces es factible comparar sus resultados para el periodo de validación.

Conclusión:

Se logró elaborar la estructura solución con la predicción *multi-step* de 12 pasos para delante de una red neuronal, en la Tabla 4 se muestra la estructura solución posee 5 entradas (MM3, M-1, M-3, M-4, M-12 y 5 neuronas en la capa oculta. Respecto de los modelos *MLPR, GBR, RFR, KNNR* se puede observar en la Tabla 7 los mejores parámetros para cada modelo. Se concluye aceptar la hipótesis alterna.

- Ho: No es posible evaluar el modelo entrenado en el período de prueba entonces no es posible generar las predicciones.

  Ha: Si es posible evaluar el modelo entrenado en el período de prueba entonces es posible generar las predicciones.

  Conclusión:

  Según la Tabla 9 se puede apreciar los resultados cuantitativos de comparación de la métrica MAPE año 2023 como resultado del modelo entrenado de enero de 2000 a diciembre de 2022, su validación de enero de 2000 a diciembre de 2022 y la prueba o test de enero a diciembre de 2023. Por lo tanto, se concluye que se acepta la hipótesis alterna.

- Ho: No hay diferencia significativa entre los algoritmos en términos de eficiencia en la predicción.

  Ha: Existe al menos una diferencia significativa entre los algoritmos en términos de eficiencia en la predicción.

Tabla 11

Resumen de prueba de hipótesis

	Hipótesis nula	Prueba	Sig.	Decisión
1	La distribución de Predicción es la misma entre las categorías de Algoritmo.	Prueba de *Kruscal-Wallis* para muestras independientes	0.005	Rechazar la hipótesis nula.

Se muestran significaciones asintóticas. El nivel de significación es de 0.005

Como el p-value = 0.005 y es menor a 0.05 entonces se rechaza la hipótesis nula y se acepta la hipótesis alterna, con lo cual se concluye que existe al menos una diferencia significativa entre los algoritmos en términos de eficiencia en la predicción. Luego se analiza esas diferencias tomando pares de algoritmos.

Figura 9

Prueba de Kruskal-Wallis para muestras independientes

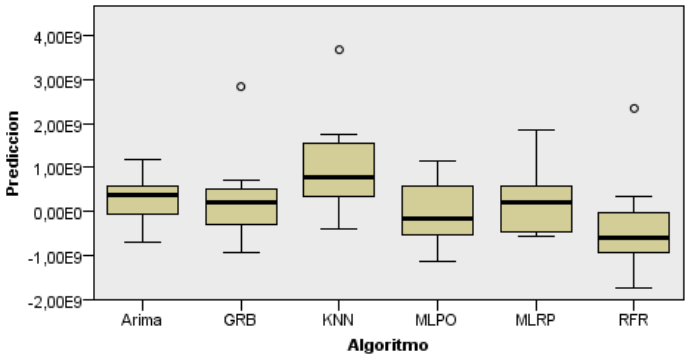


Tabla 12

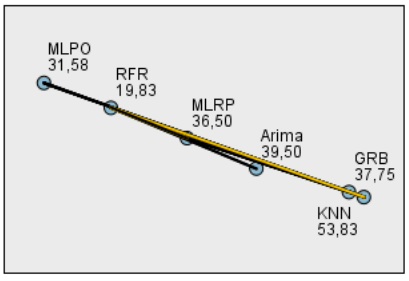
Prueba de Kruskal-Wallis para muestras independientes

N total	72
Estadístico de contraste	16.793
Grados de libertad	5
Sig. Asintótica (prueba bilateral)	0.005

1. Las estadísticas de prueba se ajustan para empates

Figura 10

Comparación entre parejas de algoritmos – gráfico de nodos



Cada nodo muestra el rango promedio de muestras de Algoritmo.

Tabla 13

Comparación entre parejas de algoritmos – gráfico de nodos

Muestra 1 – Muestra 2	Estadístico de contraste	Error	Desv. Estadístico de contraste	Sig.	Sig. Ajust.
RFR-MLPO	11.750	8.544	1.375	0.169	1.000
RFR-MLRP	16.667	8.544	1.951	0.051	0.766
RFR-GRB	17.917	8.544	2.097	0.036	0.540
RFR-Arima	19.667	8.544	2.302	0.021	0.320
RFR-KNN	34.000	8.544	3.979	0.000	0.001
MLPO-MLRP	-4.917	8.544	-0.575	0.565	1.000
MLPO-GRB	6.167	8.544	0.722	0.470	1.000
MLPO-Arima	7.917	8.544	0.927	0.354	1.000
MLRP-KNN	22.250	8.544	2.604	0.009	0.138
MLRP-GRB	1.250	8.544	0.146	0.884	1.000
MLRP-Arima	3.000	8.544	0.351	0.725	1.000
MLRP-KNN	17.333	8.544	2.029	0.042	0.637
GRB-Arima	1.750	8.544	0.205	0.838	1.000
GRB-KNN	-16.083	8.544	-1.882	0.060	0.897
Arima-KNN	-14.333	8.544	-1.678	0.093	1.000

Cada fila prueba la hipótesis nula de que las distribuciones de la Muestra 1 y la Muestra 2 son las mismas.
Se muestran las significaciones asintóticas (pruebas bilaterales). El nivel de significación es 0.005.
Los valores de significación se han ajustado mediante la corrección de Bonferroni para varias pruebas.

Del diagrama de cajas podemos realizar un primer análisis donde se muestra que el algoritmo *RFR* tienen una menor media, esto indica que es más preciso en la predicción respecto a los demás algoritmos.

De los resultados de la comparación entre parejas de algoritmos podemos observar que solo existe diferencia significativa entre la comparación de algoritmos *RFR-KNN*, indicando que el primero es más preciso para realizar la predicción.

En el gráfico de nodos se puede observar la eficiencia de predicción de cada algoritmo, se indica el que tiene menor valor medio es más preciso para realizar la predicción.

V. DISCUSIÓN DE RESULTADOS

Los resultados de los experimentos con MLP-O (*Multi-Layer Perceptron con optimización*) se ha alcanzado un Error Absoluto Medio Porcentual (MAPE) de 3.3830; *Autoregressive Integrated Moving Average (Auto-Arima)* con 3.0691 % de MAPE; Multi-*Layer Perceptron Regressor (MLPR)* de 3.5923 %; *Gradient Boosting Regressor (GBR)* de 3.6660; *Random Forest Regressor (RFR)* de 5.3389 % y 6.4466 % obtenido con el algoritmo de *K-Nearest Neighbors Regressor (KNNR)*. En la tesis de Mamani Ticona (2013) se estudia los impuestos y tasas adeudados por los contribuyentes, recibe y controlan las recaudaciones, deudas públicas y realizan los pagos de los compromisos del municipio, lo que permite planificar sus inversiones, para este efecto desarrollaron un modelo de previsión fiscal con aplicación de técnicas inteligentes – redes neuronales, logrando desarrollar la mejor arquitectura de red (MM12 y M-1 entradas y 4 neuronas) se obtuvo combinando los obtenida combinando los métodos *PCAM-ReliefF*, con MAPE igual a 5.63 % siendo el más óptimo. Con ambos resultados, el MAPE obtenido en el presente trabajo de investigación presenta un mejor valor.

Habiendo realizado el análisis de la Prueba de *Kruskal – Wallis* y comparaciones entre parejas de algoritmos el algoritmo RFR tienen una menor media, esto indica que es más preciso en la predicción respecto a los demás algoritmos, sin embargo, el algoritmo que obtuvo MAPE optimizado fue el MLP-O (*Multi-Layer Perceptron con optimización*) combinado con el algoritmo *Hill Climbing* ha alcanzado un MAPE de 3.3830 %.

Para Bravo Cucci (2023) y Yaguas (2023) una de las fuentes principales de ingresos del Estado peruano proviene de la recaudación tributaria, por ejemplo, para costear los gastos gubernamentales o el financiamiento de la construcción de obras públicas por medio de los impuestos cobrados a las empresas y a los profesionales independientes, incluso si son alterados por variables exógenas a la realidad nacional, como efectos en la salud pública Covid-

19, por ello, es importante analizar la situación de recaudación tributaria en el Perú en diversos escenarios como los efectos de factores internos – convulsiones sociales, que provocan paralizaciones de la economía y las retracciones en la generación de mano de obra e ingresos para profesionales independientes y empresas, desaceleración económica, proyecciones de incremento de pobreza y una recaudación tributaria en declive. Sin embargo, en los últimos tiempos se ha aperturado un nuevo escenario, hoy en día las administraciones tributarias cuentan con una gran cantidad de datos disponibles en formato digital, por ejemplo, hoy emiten documentos tributarios, declaraciones de impuestos en tiempo real lo cual permite una mayor estandarización de los formatos y que facilita la recaudación de impuestos a las organizaciones tributarias, facilita un mayor almacenamiento y análisis con herramientas inteligentes, por ejemplo, en mayo de 2018 se han emitido 7 160 939 584 facturas electrónicas en la Argentina, 31 292 720 000 en Brasil, 3 068 043 039 en Chile, 4 647 491 441 en Ecuador y en el Perú fue de 3 468 894 145 facturas electrónicas según (Fundación Bill & Melinda Gates, 2020).

En el trabajo de Arciniegas Paspuel et al. (2021) sobre las predicciones de recaudación tributaria de los años 2016 a 2020, mediante la metodología de Box Jenkins técnica estadística utilizada para la modelización y el análisis de series temporales determinaron al modelo ARIMA con un MAPE de 1. 52001 %.

Según De Azevedo et al. (2017) el impuesto sobre circulación de mercaderías y servicios de transporte (ICMS), es la principal fuente de ingresos por impuestos de las Unidades Federativas brasileñas, que permite provisionar su recaudación para la gestión financiera de estas entidades, a fin de mejorar esta previsión utilizaron los modelos basados en series temporales ARIMA, con un horizonte temporal de 1995 a 2013, sugieren los autores que la utilización del ARIMA efectivamente ha aumentado la precisión de las previsiones de recaudación del ICMS para 06 Estados, con un MAPE de 4.82 % para el Estado de São Paulo; Minas Gerais 15.19 %; Rio de Janeiro 7.63 %; Rio Grande do Sul 15.40 %; Paraná 222.22 %;

Bahía 27.55%, el Estado de São Paulo obtuvo estimaciones muy próximas a lo realizado en 2012 y 2013.

Delgado (2023) realizaron la caracterización del crecimiento anual de ingresos por impuesto predial los municipios mediante metodología CRISP-DM, utilizaron los algoritmos de aprendizaje automático, PCA (Análisis de componentes principales), *XGBOOST, Random Forest, RFE y SelectKBest,* con un MAPE de 20 % del modelo de regresión y clasificación XGBoost en el período de 2008 y 2020.

VI. CONCLUSIONES

- El poder de predicción del algoritmo propuesto en la presente investigación, basado en un modelo de redes *MLP* y algoritmo *Hill Climbing* con datos de la serie temporal de la SUNAT de los años 2000 a 2023 fueron eficientes, se obtuvo un error MAPE de 3.383 %; la predicción de la recaudación del año 2023 fue de 192 696 238 616.60, en comparación con el valor recaudado del año 2023 es de 192 128 881 908.62, que demuestra la eficacia del algoritmo, respecto del modelo estadístico AutoARIMA este obtuvo un MAPE de 3.0691 % habiendo una diferencia de 0.3139 %.
- Se pudo obtener la serie temporal visualizándose en la Figura 4 la trayectoria de los ingresos recaudados por la SUNAT 2000 a 2023 divididos en tres bloques: entrenamiento, validación y prueba, respecto del procesamiento se aplicó la normalización de valores de la serie en un rango de -1 a 1, logrando construir las 12 variables de entrada de las redes - Tabla 3, se ejecutó 1000 experimentos, 25 número de fallas en el conjunto de validación, se utilizó la función *tansing (Hyperbolic Tangent Sigmoid)* en la capa oculta.
- Se logró elaborar la estructura solución con la predicción *multi-step* de 12 pasos para delante de una red neuronal - Tabla 4, se muestra la estructura solución posee 5 entradas (MM3, M-1, M-3, M-4, M-12 y 5 neuronas en la capa oculta. Respecto de los modelos *MLPR, GBR, RFR, KNNR* se pudo obtener los mejores parámetros para cada modelo y sus respectivos MAPEs (Tabla 7, Tabla 8 y Tabla 9).
- Se pudo evaluar el modelo entrenado *(MLP - O)* a través de los resultados cuantitativos de comparación con la métrica MAPE para el año 2023 como resultado del modelo entrenado de enero de 2000 a diciembre de 2022, su validación de enero de 2000 a diciembre de 2022 y la prueba o test de enero a diciembre de

2023. Las predicciones de la técnica *hyperparameter tuning,* presentaron resultados muy próximos comparado con la técnica basada en modelos estadísticos, Auto-ARIMA, la diferencia es menos significativa entre los seis algoritmos en términos de eficiencia.

- Existe al menos una diferencia significativa entre los seis modelos en términos de eficiencia en la predicción, el algoritmo *RFR* tienen una menor media, esto indica que es más preciso en la predicción respecto a los demás algoritmos. No obstante, el resultado presentado por nuestro modelo *(MLP-O)* muestra un resultado muy próximo al mejor (Auto-ARIMA), haciendo una comparación específica del modelo desarrollado en la tesis *(MLP-O)* con el modelo MLPR, nuestro modelo se muestra superior, esto ocurre en razón a la inclusión del método *Hill Climbing*

VII. RECOMENDACIONES

- Como perspectiva futura, se pretende extender el modelo para predecir otras series tributarias y utilizar otros modelos híbridos como por ejemplo redes recurrentes LSTM *(Long Short-Term Memory)*, así pues, los resultados sugieren que la adopción de la metodología de previsión ARIMA podría ser utilizada para aumentar la previsibilidad de la recaudación, aunque deben tenerse en cuenta las críticas que se han hecho al uso de esta metodología para la previsión de ingresos, en la literatura existente, se presentan proyecciones de series temporales para los ingresos públicos no parecen ser muy efectivos para que las instituciones públicas en este caso el Ministerio de Economía y Finanzas pueda usar en sus previsiones futuras, ya que se limitan a desarrollar un modelo ideal, poniendo en riesgo las estimaciones futuras, por ello es recomendable que las universidades formen especialistas en ciencia de datos, inteligencia artificial, para desarrollar estas metodologías.
- Complementariamente en base a la investigación realizada se recomendaría emplear ensambles para combinar los 3 primeros modelos que presentan mejores resultados. De esta forma, se combinaría el aprendizaje de los 3 modelos para verificar posibles mejoras en el desempeño de predicción de las series temporales de los ingresos de la SUNAT, con este modelo propuesto para la recaudación de impuestos pueda utilizarse también para mejorar (reducir el error MAPE) la calidad de las previsiones de otros ingresos como en minería, hidrocarburos, pesca, agropecuario, como también, disponer la información histórica en la base de datos de SUNAT.

- Se sugiere que para la estimación de los ingresos públicos recaudados puedan utilizar técnicas econométricas para analizar el ajuste con modelos estadísticos, redes neuronales, etc., para mejorar el proceso de toma de decisiones, ahondando la exploración utilizando procesos para demostrar causalidad.

VIII. REFERENCIAS

Al-Betar, M. A., Awadallah, M. A., Makhadmeh, S. N., Doush, I. A., Zitar, R. A., Alshathri, S., & Abd Elaziz, M. (2023). A hybrid Harris Hawks optimizer for economic load dispatch problems. *Alexandria Engineering Journal*, *64*, 365–389. https://doi.org/10.1016/j.aej.2022.09.010

Alonso Ramirez Gil, W., & Ramirez Gil, C. M. (2023). *Programacion de Inteligencia Artificial: curso practico*. RA-MA Editorial. elibro.bibliotecaupn.elogim.com/es/lc/upnorte/titulos/235051

Arciniegas, O. G., Castro, L. G., & Arias, W. M. (2021). Análisis y predicción de la recaudación tributaria en el Ecuador ante la COVID-19, aplicando el modelo ARIMA. *Dilemas Contemporáneos: Educación, Política y Valores*. https://doi.org/10.46377/dilemas.v8i.2708

Arciniegas Paspuel, O. G., Castro Morales, L. G., & Arias Collaguazo, W. M. (2021). Análisis y predicción de la recaudación tributaria en el Ecuador ante la COVID-19, aplicando el modelo ARIMA. *Dilemas Contemp. Educ. Política Valores*, *8*. https://doi.org/https://doi.org/10.46377/dilemas.v8i.2708

Bastos, & Campos, G. A. L. (2010). *Forecast of tax revenues through artificial neural networks*. Ceará State University.

Benitez Iglesias, R. (2014). *Inteligencia artificial avanzada*. Editorial UOC. elibro.bibliotecaupn.elogim.com/es/lc/upnorte/titulos/57582

Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for Hyper-Parameter Optimization. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems* (Vol. 24). Curran Associates, Inc.

https://proceedings.neurips.cc/paper_files/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, *13*(2).

Bhattacharya, A., Saha, B., Chattopadhyay, S., & Sarkar, R. (2023). Deep feature selection using adaptive -Hill Climbing aided whale optimization algorithm for lung and colon cancer detection. *Biomedical Signal Processing and Control*, *83*, 104692. https://doi.org/10.1016/j.bspc.2023.104692

Booba, B., Joshphin Jasaline Anitha, X., Mohan, C., & S, J. (2024). Hybrid approach for virtual machine allocation in cloud computing. *Sustainable Computing: Informatics and Systems*, *41*, 100922. https://doi.org/10.1016/j.suscom.2023.100922

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time Series Analysis: Forecasting and Control* (5th Editio).

Bravo Cucci, J. (2023, April 11). *Panorama de la recaudación tributaria en el Perú*. Conexiónesan. https://www.esan.edu.pe/conexion-esan/panorama-de-la-recaudacion-tributaria-en-el-peru

Cabrera, M., Sánchez-Chero, M., Cachay, L., & Rosas-Prado, C. (2021). *Cultura tributaria y su relación con la evasión fiscal en Perú*. *27*(3), 204–218. https://dialnet.unirioja.es/servlet/articulo?codigo=8081767

Cáceres Hernández, J. J., Martin Rodriguez, G., & Martin Alvarez, F. J. (2007). *Introducción al Análisis Univariante de series temporales económicas*. Delta Publicaciones. elibro.bibliotecaupn.elogim.com/es/lc/upnorte/titulos/170132

Caro Arroyo, J. M. (2020). Los modelos de tributación en Latinoamérica y su incidencia en la desigualdad. *Revista Científica General José María Córdova*, *18*(31), 675–706. https://doi.org/10.21830/19006586.583

Castillo-Reyes, G., Estrella, R., Roose, D., Abrams, F., Jiménez-Moya, G., & Van Orshoven, J. (2024). Spatially targeted afforestation to minimize sediment loss from a catchment: An efficient hill climbing method considering spatial interaction. *Environmental Modelling & Software*, *176*, 106000. https://doi.org/10.1016/j.envsoft.2024.106000

Celikay, F. (2020). Dimensions of tax burden: a review on OECD countries. *Journal of Economics, Finance and Administrative Science*, *25*(49), 27–43. https://doi.org/10.1108/JEFAS-12-2018-0138

Coaquira Taboada, J. N., Chaupis Pajuelo, D. A., & Burgos Zavaleta, V. F. J. (2022). Ingresos tributarios provenientes del sector minero y su relación con la recuperación económica del Perú. *Quipukamayoc*, *30*(63), 59–68. https://doi.org/10.15381/quipu.v30i63.23327

Comexperu. (2024, March 1). *Radiografía del Sector Informal: ¿Por qué se perdieron más de 600 000 empleos informales en el 2023?* .

De Azevedo, R. R., da Silva, J. M., & Gatsios, R. C. (2017). Analise critica dos modelos de previsao de serie temporal com base no ICMS estadual. *Revista De Gestao, Financas E Contabilidade*, *7*, 164+. https://repositorio.usp.br/item/002881180

De Castro Felippetto, M. C. (2001). *Predição Não-Linear de Séries Temporais Usando Redes Neurais RBF por Decomposição em Componentes Principais* [Universidade Estadual de Campinas - UNICAMP]. https://www.fccdecastro.com.br/pdf/CristinaThesis.pdf

De La Cruz Casaño, C. (2016). Metodología de la investigación tecnológica en ingeniería. *Ingenium*, *1*(1). https://journals.continental.edu.pe/index.php/ingenium/article/view/392

Del Carpio Gallegos, J. (2005). Las redes neuronales artificiales en las finanzas. *Revista de La Facultad de Ingeniería Industrial*, *8*(2), 28–32. https://sisbib.unmsm.edu.pe/bibvirtualdata/publicaciones/indata/Vol8_n2/a05.pdf

Delgado, D. (2023). *Caracterización de los municipios de Colombia según el potencial de crecimiento del ingreso por impuesto predial utilizando Machine Learning* [Universidad

de La Sabana]. https://intellectum.unisabana.edu.co/bitstream/handle/10818/59338/Proyecto_Grado_John_Daniel_Delgado_Biblioteca.pdf?sequence=1&isAllowed=y

Desfrancois, P. (2023). Determinantes de la transparencia fiscal en América Latina. *RHS-Revista Humanismo y Sociedad*, *11*(2). https://doi.org/10.22209/rhs.v11n2a04

Escuela de Administración de Negocios para Graduados [ESAN]. (2024). *Conexioesan*. https://www.esan.edu.pe/conexion-esan/economia-peruana-informalidad-aumentaria-hasta-en-un-78-en-el-2024#:~:text=en el 2024-,Economía peruana%3A informalidad aumentaría hasta en un 78%25 en el,de pobreza para el 2024.

Espinoza, Á., Fort, R., & Espinoza, M. (2022). *El impacto de la pandemia en el sistema de distribución de alimentos del Perú: los mercados de abastos minoristas* (126th ed.). Grupo de Análisis para el Desarrollo (GRADE). https://www.grade.org.pe/en/publicaciones/el-impacto-de-la-pandemia-en-el-sistema-de-distribucion-de-alimentos-del-peru-los-mercados-de-abastos-minoristas/

Faceli, K., Lorena, A. C., Gama, J., & Carvalho, A. C. P. de L. F. de. (2011). *Inteligência Artificial. Uma Abordagem de Aprendizado de Máquina*. LTC.

Fernández-García, P., Vallejo-Seco, G., Livacic-Rojas, P. E., & Tuero-Herrero, E. (2014). Validez Estructurada para una investigación cuasi-experimental de calidad: se cumplen 50 años de la presentación en sociedad de los diseños cuasi-experimentales. *Anales de Psicología*, *30*, 756–771.

Fierro, M. (2011). El desarrollo conceptual de la ciencia cognitiva. Parte I. *Revista Colombiana de Psiquiatría*, *40*(3), 519–533. https://doi.org/10.1016/S0034-7450(14)60144-X

Flores, N. (2022). *Estrategias de cultura tributaria para estudiantes de secundaria del área rural, para un futuro cumplimiento de obligaciones tributarias* [Trabajo de grado, Universidad Mayor de San Simón, Bolivia].

http://ddigital.umss.edu.bo:8080/jspui/handle/123456789/30338

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, *29*(5). https://doi.org/10.1214/aos/1013203451

Fundación Bill & Melinda Gates. (2020). *Las TIC como herramienta estratégica para potencial la eficiencia de las administraciones tributarias*. Centro Interamericano de Administraciones Tributarias – CIAT. https://www.ciat.org/Biblioteca/Estudios/2020_TIC-CIAT-FBMG.pdf

García Diaz, J. C. (2016). *Predicción en el dominio del tiempo: análisis de series temporales para ingenieros*. Universidad Politecnica de Valencia. elibro.bibliotecaupn.elogim.com/es/lc/upnorte/titulos/57439

Garcia Jimenez, M. V. (2014). *Disenos experimentales de series temporales*. UNED - Universidad Nacional de Educacion a Distancia. elibro.bibliotecaupn.elogim.com/es/lc/upnorte/titulos/48742

Ghawi, R., & Pfeffer, J. (2019). Efficient Hyperparameter Tuning with Grid Search for Text Categorization using kNN Approach with BM25 Similarity. *Open Computer Science*, *9*(1), 160–180. https://doi.org/10.1515/comp-2019-0011

Gutierrez Portela, F., Rodríguez Cárdenas, S., Patiño Ospina, L. P., & Hernandez Aros, L. (2023). Estudio de la prevención y detección de fraudes financieros a través de técnicas de aprendizaje automático. *CAFI*, *6*(1), 77–101. https://doi.org/10.23925/cafi.v6i1.58372

Hernández Aros, L., Gutierrez Portela, F., & Rodriguez Tovar, K. L. (2023). Análisis del uso de técnicas supervisadas de aprendizaje automático y profundo en la detección de fraude financiero. In *Revista Tecnologia en Marcha*. El Instituto. https://repository.ucc.edu.co/entities/publication/8b2b8fe9-3a73-4206-87a8-d57097af3a45

Hernández Márquez, V. Y. (2018). *Generación de trayectorias para tareas de navegación*

autónoma y mapeo en ambiente sinteriores [Trabajo de grado, Universidad Politécnica de Tulancingo]. https://www.upt.edu.mx/Contenido/Investigacion/Contenido/TESIS/MAC/2018/MAC_T_2018_01_VHM.pdf

Herrera Maguiña, D. E. (2021). *La cultura tributaria efectiva y la reducción de la defraudación tributaria en la Intendencia Lima de la SUNAT* [Tesis de grado, Universidad Nacional Federico Villarreal]. https://hdl.handle.net/20.500.13084/5101

Hinton, G. E. (1989). Connectionist learning procedures. *Artificial Intelligence*, *40*(1–3), 185–234. https://doi.org/10.1016/0004-3702(89)90049-0

Instituto Peruano de Economía. (2020). Impacto del COVID-19 en Perú y Latinoamérica . In *Boletín de discusión* . Instituto Peruano de Economía. ttps://www.ipe.org.pe/portal/wp-content/uploads/2020/10/2020-10-15-boletín-impacto-del-coivd-19-en-perú-y-latinomaerica.pdf

Ioniţă, A., Banu, D.-A., & Oleniuc, I. (2023). Heuristic Optimizations of Boolean Circuits with Application in Attribute-Based Encryption. *Procedia Computer Science*, *225*, 3173–3182. https://doi.org/10.1016/j.procs.2023.10.311

Kaldor, N. (2021). El papel de la tributación en el desarrollo económico. *El Trimestre Económico*, *88*(352), 1215–1244. https://doi.org/10.20430/ete.v88i352.1346

Kim, S., & Kim, H. (2016). A new metric of absolute percentage error for intermittent demand forecasts. *International Journal of Forecasting*, *32*(3), 669–679. https://doi.org/10.1016/j.ijforecast.2015.12.003

Kingma, D. P., & Lei Ba, J. (2015). *Adam: A Method for Stochastic Optimization*. 1–15.

Lajunen, A. (2014). Fuel economy analysis of conventional and hybrid heavy vehicle combinations over real-world operating routes. *Transportation Research Part D: Transport and Environment*, *31*, 70–84. https://doi.org/10.1016/j.trd.2014.05.023

Lauletta, M. ., & Montaño Campos, F. (2018). Government pardons and tax compliance: the importance of wealth and access to public goods. *Revista de Economía Política de Buenos Aires*, *12*(17), 185–205. https://ojs.econ.uba.ar/index.php/REPBA/article/view/1324

Ludeña Dávila, M. K., & Tonon Ordóñez, L. B. (2021). Calculando el riesgo de insolvencia, de los métodos tradicionales a las redes neuronales artificiales. Una revisión de literatura . *INNOVA Research Journal*, *6*(3), 270–287. https://dialnet.unirioja.es/servlet/articulo?codigo=8226196

Mamani, W. (2013). *Estudo de Métodos de Mineração de Dados Aplicados à Gestão Fazendária de Municípios*. Pontifícia Universidade Católica do Rio de Janeiro.

Mamani, W., Figueiredo, K. T., & Rebuzzi, M. M. B. (2017). Predicción multi-step del impuesto sobre servicios usando redes neuronales artificiales y métodos de selección de variables. *Revista Científica Andina - Science & Humanities*, 77–90.

Mansour, Y., & Schain, M. (2001). Learning with Maximum-Entropy Distributions. *Machine Learning*, *45*(2), 123–145. https://doi.org/10.1023/A:1010950718922

Mendelsohn, L. B. ·. (2000). *Trend Forecasting with Technical Analysis: Unleashing the Hidden Power of Intermarket Analysis to Beat the Market*. Marketplace Books, Incorporated.

Meseguer Gonzalez, P., & Lopez de Mantaras Badia, R. (2017). *Inteligencia artificial*. Editorial CSIC Consejo Superior de Investigaciones Cientificas. elibro.bibliotecaupn.elogim.com/es/lc/upnorte/titulos/42319

Molina-Muñoz, J. (2021). Análisis bibliométrico del uso de Machine Learning en finanzas a través de un modelo K-Means. *Eficiencia*, *1*(3). https://doi.org/https://doi.org/10.15765/ys0fp136

Morales España, G., Mora Flórez, J., & Vargas Torres, H. (2008). Estrategia de regresión basada en el método de los k vecinos más cercanos para la estimación de la distancia de

falla en sistemas radiales. *Rev. Fac. Ing. Univ. Antioquia* , *45*, 100–108. http://www.scielo.org.co/pdf/rfiua/n45/n45a09.pdf

Morales Millan, A. V., & Cely García, F. E. (2024). *Descripción de las causas y consecuencias de la evasión fiscal en los tributos del departamento de Boyacá durante el periodo 2020-2023* [Trabajo de grado, Universidad Santo Tomás.]. https://repository.usta.edu.co/handle/11634/55191

Moreno Kong, J. M. (2018). *Influencia del PBI y la inflación en el ingreso tributario del Perú, periodo 2003-2017* [Tesis de grado, Universidad Privada Antenor Orrego]. https://repositorio.upao.edu.pe/handle/20.500.12759/4352

Müller, A., & Guido, S. (2017). *Introduction to Machine Learning with Python. A Guide for Data Scientists*. O'Reilly Media.

Naskar, A., Pramanik, R., Hossain, S. K. S., Mirjalili, S., & Sarkar, R. (2023). Late acceptance hill climbing aided chaotic harmony search for feature selection: An empirical analysis on medical data. *Expert Systems with Applications*, *221*, 119745. https://doi.org/10.1016/j.eswa.2023.119745

Night, S., & Bananuka, J. (2019). The mediating role of adoption of an electronic tax system in the relationship between attitude towards electronic tax system and tax compliance. *Journal of Economics, Finance and Administrative Science*, *25*(49), 73–88. https://doi.org/10.1108/JEFAS-07-2018-0066

Oddi, F. J., Aristimuño, F. J., Coulin, C., & Garibaldi, L. A. (2018). Ambigüedades en términos científicos: El uso del "error" y el "sesgo" en estadística. *Ecología Austral*, *28*(3), 525–536. https://doi.org/10.25260/EA.18.28.3.0.680

Palma Méndez, J., & Marín Morales, R. (2008). *Inteligencia Artificial. Técnicas, métodos y aplicaciones*. Mc Graw Hill. https://ebooks724.bibliotecaupn.elogim.com:443/?il=7368

Pérez Valqui, C. M. (2018). *La cultura tributaria como herramienta eficaz para reducir la*

evasión fiscal en los contribuyentes del impuesto a la renta de tercera categoría de Lima Metropolitana [Tesis de grado, Universidad Nacional Federico Villarreal]. https://repositorio.unfv.edu.pe/handle/20.500.13084/2066

Piancastelli, M., & Thirlwall, A. P. (2020). The Determinants of Tax Revenue and Tax Effort in Developed and Developing Countries: Theory and New Evidence 1996-2015. *Nova Economia*, *30*(3), 871–892. https://doi.org/10.1590/0103-6351/5788

Qi, X., Gao, Y., Li, Y., & Li, M. (2022). K-nearest Neighbors Regressor for Traffic Prediction of Rental Bikes. *2022 14th International Conference on Computer Research and Development (ICCRD)*, 152–156. https://doi.org/10.1109/ICCRD54409.2022.9730527

Ramírez, J. (2020). Metodología Box-Jenkins para la estimación de la demanda de Habanos en México. *Revista Dilemas Contemporáneos: Educación, Política y Valores, 8*(28), 1–5.

Rodriguez-Tovar, K. L., Gutiérrez-Portela, F., & Hernández-Aros, L. (2023). Análisis del uso de técnicas supervisadas de aprendizaje automático y profundo en la detección de fraude financiero . *Tecnología En Marcha*, *36*(8), 50–56. https://dialnet.unirioja.es/servlet/articulo?codigo=9270442

Romero-Carazas, R., Chambilla Choquecahua, M., Santivañez Villavicencio, Y. M., Santos Maldonado, A. B., & Ugarte Portuondo, W. A. (2022). La cultura y las obligaciones tributarias en una empresa peruana. *Ciencia Latina Revista Científica Multidisciplinar*, *6*(4), 3279–3292. https://doi.org/10.37811/cl_rcm.v6i4.2833

Santiago, A. M., Vanstrahlengs, J. U., Otero, A. C., & Lombana, J. (2017). Pronóstico del precio de la energía en Colombia utilizando modelos ARIMA con IGARCH. *Revista de Economía Del Rosario*, *20*(1), 127–159.

SKTIME. (2021). *AutoARIMA*. https://www.sktime.net/en/v0.19.0/api_reference/auto_generated/sktime.forecasting.arima.AutoARIMA.html

SKTIME. (2024). *A unified framework for machine learning with time series.* https://www.sktime.net/en/stable/

Superintendencia Nacional de Administración Tributaria [SUNAT]. (2024). *Estadísticas y estudios*. Nota Tributaria y Aduanera. https://www.sunat.gob.pe/estadisticasestudios/ingresos-recaudados.html

Supo, J. (2023). *Niveles de investigación*. Bioestadístico. https://bioestadistico.com/niveles-de-investigacion

Syed, D., Refaat, S. S., & Abu-Rub, H. (2020). Performance Evaluation of Distributed Machine Learning for Load Forecasting in Smart Grids. *2020 Cybernetics & Informatics (K&I)*, 1–6. https://doi.org/10.1109/KI48306.2020.9039797

Viniegra Velázquez, L. (2014). El reduccionismo científico y el control de las conciencias: *Boletín Médico Del Hospital Infantil de México*, *71*(4), 252–257. https://doi.org/10.1016/j.bmhimx.2014.05.001

Wang, J., Zhou, G., Lin, D., Hong, Y., Liang, Z., Dong, R., & Yang, L. (2023). An autofocusing method for dynamic surface-enhanced Raman spectroscopy detection realized by optimized hill-climbing algorithm with long time stable hotspots. *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, *299*, 122820. https://doi.org/10.1016/j.saa.2023.122820

Yaguas, E. (2023, October 15). *Economía peruana en rojo: recaudación tributaria a la baja y millonarias deudas pendientes*. Ojo Público. https://ojo-publico.com/4713/economia-rojo-menos-recaudacion-y-millonarias-deudas-tributarias

Zainab, A., Ghrayeb, A., Houchati, M., Refaat, S. S., & Abu-Rub, H. (2020). Performance Evaluation of Tree-based Models for Big Data Load Forecasting using Randomized Hyperparameter Tuning. *2020 IEEE International Conference on Big Data (Big Data)*, 5332–5339. https://doi.org/10.1109/BigData50022.2020.9378423

Zhang, Q., Tang, D., Quan, Q., Jin, Y., & Deng, Z. (2023). Hill-climbing & fuzzy combined control algorithm for a percussive ultrasonic drill. *Applied Acoustics*, *211*, 109499. https://doi.org/10.1016/j.apacoust.2023.109499

ANEXO 1

INGRESOS RECAUDADOS POR LA SUNAT, 2000 - 2023 (Millones de Soles)

AÑO	ENERO	FEBRERO	MARZO	ABRIL	MAYO	JUNIO	JULIO	AGOSTO	SEPTIEMBRE	OCTUBRE	NOVIEMBRE	DICIEMBRE	TOTAL
2000	2,700.6	2,243.4	2,321.1	2,659.9	2,424.4	2,250.8	2,329.2	2,508.2	2,257.1	2,349.9	2,351.0	2,440.0	28,835.3
2001	2,684.7	2,468.8	2,384.5	2,870.1	2,493.8	2,294.5	2,475.5	2,506.8	2,370.6	2,358.6	2,441.0	2,425.6	29,774.6
2002	2,635.8	2,086.8	2,061.0	2,969.3	2,390.2	2,451.1	2,487.9	2,687.3	2,680.8	2,580.8	2,630.4	2,776.6	30,437.8
2003	2,913.8	2,478.4	2,897.8	3,142.6	2,726.7	2,699.9	2,699.4	2,979.5	2,789.6	2,809.7	2,934.5	3,106.8	34,178.7
2004	3,178.7	2,781.6	3,118.1	3,532.6	3,216.5	3,226.6	3,093.5	3,309.9	3,266.0	3,140.5	3,156.6	3,423.6	38,444.2
2005	3,548.4	3,041.2	3,684.3	4,999.8	3,502.2	3,475.6	3,286.1	3,699.2	3,686.9	3,532.0	3,670.3	3,885.7	44,011.7
2006	4,690.8	3,532.2	4,335.5	6,729.2	4,328.6	4,281.7	4,986.8	4,513.5	4,214.8	4,492.1	4,613.2	4,788.2	55,506.5
2007	5,193.7	4,113.0	4,935.6	8,276.9	4,861.0	4,935.7	5,283.0	5,548.2	5,053.1	5,387.9	5,470.4	5,261.1	64,319.5
2008	6,261.8	4,931.6	5,568.1	7,837.0	5,649.9	5,908.7	6,513.9	6,157.5	6,212.7	6,277.6	5,552.4	5,889.1	72,760.4
2009	6,237.1	4,928.7	5,653.9	7,103.4	5,062.2	5,198.5	5,196.8	5,512.6	5,272.4	5,638.6	5,785.3	6,010.4	67,599.9
2010	6,983.5	5,617.1	6,764.3	8,767.4	6,083.7	6,173.1	6,529.9	6,674.1	6,492.0	6,840.3	6,996.7	7,159.0	81,081.0
2011	8,063.1	6,659.9	8,387.0	10,320.5	7,606.2	7,461.5	7,423.4	7,952.3	7,807.8	7,803.9	7,840.5	7,956.8	95,282.9
2012	9,571.2	7,438.0	9,462.0	11,126.8	8,539.3	8,085.6	8,784.9	8,183.0	8,632.3	8,669.8	8,987.9	9,010.5	106,494.0
2013	9,846.4	8,288.0	9,759.0	11,010.3	9,033.9	8,510.2	9,100.2	9,265.9	9,427.9	9,326.7	9,590.4	10,178.9	113,337.7
2014	10,725.4	8,668.6	11,019.4	11,752.4	8,932.0	8,908.7	8,902.6	10,557.2	9,543.6	9,449.9	10,708.0	10,037.3	119,205.1
2015	10,630.1	8,774.3	10,332.9	10,930.2	8,755.7	9,005.0	9,097.7	9,613.8	9,376.2	9,306.2	9,537.0	10,211.9	115,571.0
2016	10,683.8	9,028.4	11,096.4	12,181.1	9,119.2	9,273.4	8,953.8	9,688.1	9,678.0	9,518.1	9,558.8	11,017.1	119,796.2
2017	11,000.1	9,061.0	10,671.4	11,453.8	9,447.2	9,196.5	9,596.4	9,982.2	9,735.2	10,303.3	10,365.2	11,955.6	122,767.7
2018	11,777.8	9,950.6	11,877.4	15,162.8	10,864.2	10,864.0	10,961.2	10,756.5	11,083.0	11,050.2	11,286.4	11,803.5	137,437.5
2019	12,928.9	10,788.0	12,224.6	15,886.9	12,007.7	11,245.2	11,011.9	11,228.2	11,946.3	11,941.5	12,251.2	12,376.5	145,836.7
2020	13,754.6	10,960.3	10,482.3	10,289.5	9,532.0	7,041.9	9,042.3	9,761.3	10,130.8	11,066.7	11,626.6	12,553.4	126,241.7
2021	13,548.3	12,255.4	14,956.2	17,890.5	13,191.4	13,344.7	13,749.4	18,187.6	14,578.5	14,334.6	16,491.4	17,474.3	180,002.1
2022	16,419.7	14,140.0	21,398.0	24,888.3	16,350.6	14,880.5	14,556.1	16,708.8	16,327.7	16,305.6	15,826.9	15,905.6	203,708.0
2023	16,864.6	15,027.1	19,054.5	20,153.4	15,730.2	14,111.9	14,648.7	15,566.0	14,614.2	15,310.3	15,982.4	15,065.6	192,128.9

Fuente: https://www.sunat.gob.pe/estadisticasestudios/ingresos-recaudados.html

ANEXO 2

RESULTADOS DE PROYECCIONES DE INGRESOS RECAUDADOS PARA LA SUNAT PARA EL AÑO 2023 (Millones de Soles) CON OTROS MODELOS DE APRENDIZAJE SUPERVISADO Y ESTADÍSTICO

	Real	Auto-ARIMA	MLP-O	MLPR	GBR	RFR	KNN
Ene	16864577392.81	16501522678.751358	15713808589.71	15020415717.680843	16753759603.881516	17457514122.45695	16929919764.479353
Feb	15027128137.53	14712339401.06374	14589818911.03	14678501707.494717	14394850090.634031	14685427074.886217	13256777793.706667
Mar	19054534827.42	17882411978.249767	18335023994.66	18212199938.3835	16211464679.810974	16704249575.145267	15370542236.691826
Abr	20153407083.73	19585105733.228195	20853554728.77	19508420114.943928	19753684548.875507	20514540133.965477	18756856600.86002
May	15730199663.28	15160297203.458702	16342988404.86	15608235332.315584	15028023942.705778	15429929143.237532	14045620890.314665
Jun	14111888890.55	14000935646.224586	14566081810.90	14595380812.583607	14432322648.306963	15005407417.678045	13391079722.280514
Jul	14648739285.32	14287385885.955675	14774637559.09	14130729912.748768	14502400843.42526	15457904738.453281	14048779631.688557
Ago	15565972895.45	15180374404.28693	15635955644.75	16135564435.988571	15849209741.887314	16510344148.513939	15100880607.237007
Set	14614241126.18	15315518166.484749	15740290502.10	15160916946.482729	14978182208.130684	15582116069.72732	14391574879.220192
Oct	15310267710.61	15534897916.97216	15620916748.66	14998075666.645493	15047474676.693962	15903901974.745966	14480914332.269009
Nov	15982371790.94	15400569802.650326	15289594200.23	16420982139.982237	15633502399.5299	16351339354.12609	14946533361.92257
Dic	15065553104.8	15721149120.579323	15233567521.85	15413585595.839264	16011793606.49585	16799616930.15003	15458035191.533752