

# Informe Científico: Técnicas De Machine Learning Para Procesamiento De Series Temporales

Edgar Leandro Jiménez Jaimes  
Maestría Ciencias de los Datos y Analítica  
Universidad EAFIT  
eljimenezj@eafit.edu.co

Santiago Echeverri Calderón  
Maestría Ciencias de los Datos y Analítica  
Universidad EAFIT  
sechev11@eafit.edu.co

**Resumen—** Los avances en tecnologías de captura, transmisión y almacenamiento de información han promovido una intensiva generación de datos temporales en múltiples campos de la ciencia y la industria. El pronóstico de series de tiempo es una técnica importante para explotar estos datos. Resulta necesario analizar los modelos de estadísticos clásicos en comparación con métodos de aprendizaje automático para determinar qué ventajas ofrecen los unos sobre los otros.

**Keywords—** Revisión sistemática, series de tiempo, aprendizaje automático, métodos estadísticos.

## I. INTRODUCCIÓN

Los avances tecnológicos han permitido que se incluyan diferentes tipos de sensores en prácticamente todo tipo de equipos y maquinaria, que junto con la tecnología IoT y la computación en la nube han permitido la disponibilidad de datos temporales y su incremento acelerado en múltiples campos de la ciencia y la industria como el transporte, la agricultura, la medicina, los procesos productivos y las finanzas. El pronóstico de series de tiempo es una técnica importante para explotar estos datos, resulta entonces necesario analizar los modelos que mejor desempeño tengan para esta tarea. El objetivo del presente estudio es realizar una revisión de literatura, en la que se investiguen modelos estadísticos clásicos en comparación con métodos de aprendizaje automático para determinar qué ventajas ofrecen los unos sobre los otros en su aplicación sobre series temporales.

## II. MARCO TEÓRICO

Las series temporales son conjuntos de observaciones en intervalos de tiempo separados de manera regular. Pueden dividirse en series de tiempo univariadas y multivariadas, y generalmente tienen constancia de un gran tamaño de datos, alta dimensionalidad y cambio constante. Una serie de tiempo univariante, como su nombre indica, es una serie con una sola variable dependiente del tiempo. Por el contrario, una serie de tiempo multivariante tiene más de una variable dependiente del tiempo, es decir, los valores a predecir dependen de sus valores pasados y también de otras variables [1].

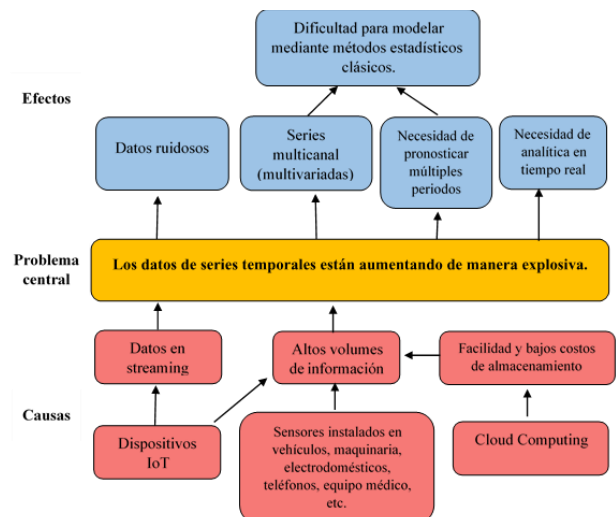
Tradicionalmente los pronósticos de series temporales se han trabajado con métodos estadísticos clásicos como ARIMA (Autoregressive Integrated Moving Average) y Suavizamiento Exponencial. Y aunque estas técnicas ofrecen un buen desempeño, el Aprendizaje Automático y el Aprendizaje Profundo están comenzando a surgir como métodos alternativos, que en otros tipos de problema y conjuntos de datos no temporales han demostrado tener muy buen desempeño.

Otro aspecto importante para considerar es para cuántos periodos futuros se requiere un pronóstico. Generalmente, el

pronóstico de series de tiempo describe la predicción de la observación en el siguiente paso de tiempo (one-step forecasting). Sin embargo, hay algunos problemas de series de tiempo en los que se deben predecir varios periodos futuros, estos pronósticos se denominan de pasos múltiples (multi-step forecasting).

## III. NECESIDAD DE LA REVISIÓN

A continuación, se presenta la necesidad en forma de árbol del problema:



## IV. METODOLOGÍA

Para la revisión sistemática de literatura se definió el siguiente protocolo de búsqueda:

### A. Preguntas de búsqueda

1. ¿Tienen un mejor desempeño los modelos de Machine Learning que los Métodos Clásicos en series de tiempo?
2. ¿Qué técnicas de Machine Learning han sido usadas con éxito en series de tiempo?

### B. Criterios de inclusión

Se incluirán en los resultados los documentos que cumplan con los siguientes criterios:

- Artículos científicos.
- Publicaciones en inglés.
- Publicaciones desde el año 2015 en adelante.

### C. Criterio de exclusión

Se excluirán en los resultados los documentos que cumplan con los siguientes criterios:

- Publicaciones no digitales.
- Publicaciones no disponibles para revisión completa.

### D. Estrategias de búsqueda – Bases de datos

La búsqueda de artículos se realizará en las siguientes bases de datos:

- Scopus
- Google Scholar

### E. Estrategias de búsqueda – Pasos de la búsqueda

La búsqueda se realizará de la siguiente forma:

- Consultar las cadenas de búsqueda en las bases de datos definidas y obtener hasta los 1,000 artículos más citados para cada cadena y base de datos.
- Consolidar los listados.
- Filtrar los resultados según los criterios de inclusión y exclusión.
- Eliminar los artículos repetidos.
- Seleccionar los 15 artículos más citados.
- Seleccionar los 2 artículos que se consideren más relevantes a partir del número de citaciones y relevancia para cada pregunta después de leer el resumen, la introducción y las conclusiones aplicando las preguntas de calidad.

### F. Términos de búsqueda

Los temas principales de la búsqueda son Aprendizaje Automático, series de tiempo y métodos clásicos. Para cada uno de estos temas se definieron los términos clave que se presentan a continuación:

Para Machine Learning:

- A1 Regression
- A2 Machine Learning
- A3 Artificial Intelligence

Para series de tiempo:

- B1 Time series
- B2 Temporal data
- B3 Forecasting
- B4 Multivariate
- B5 Techniques

Métodos clásicos

- C1 ARIMA
- C2 Classic methods
- C3 Classic models

### G. Cadenas de búsqueda

Usando los términos de búsqueda se definieron las siguientes cadenas de búsqueda para cada pregunta:

*Pregunta 1: ¿Tienen un mejor desempeño los modelos de Machine Learning que los métodos clásicos en series de tiempo?*

(A1 OR A2 OR A3) AND (B1 OR B2) AND (B3 OR B5) AND (C1 OR C2 OR C3)

*Pregunta 2: ¿Qué técnicas de Machine Learning han sido usadas con éxito en series de tiempo?*

A2 AND B4 AND (B1 OR B2) AND (B3 OR B5)

### H. Preguntas de calidad

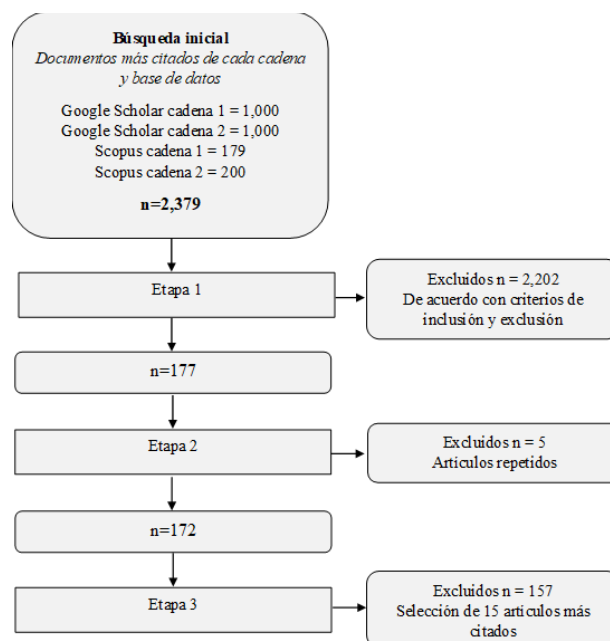
Para evaluar y escoger los artículos finales de la revisión se usarán las siguientes preguntas de calidad:

- ¿Qué técnicas de Machine Learning usan?
- ¿Cuál es la rigurosidad del análisis que hacen?
- ¿Qué métodos de validación usan para comparar las técnicas de ML vs los métodos clásicos?
- ¿Qué tipo de dataset utilizaron para probar los modelos?

## V. BÚSQUEDA DE DOCUMENTOS

La búsqueda se realizó utilizando el software *Publish or Perish*. La búsqueda en *Google Scholar* se limitó a los 1000 documentos más citados y la de *Scopus* a los 200 documentos más citados.

Esta búsqueda se realizó sin aplicar ningún tipo de filtro y después de obtener y consolidar los resultados se aplicaron los criterios de inclusión y exclusión y las preguntas de calidad como se muestra en el siguiente diagrama:



## VI. ANÁLISIS DE RESULTADOS

### A. Resultados pregunta 1: ¿Tienen un mejor desempeño los modelos de Machine Learning que los métodos clásicos en series de tiempo?

De la búsqueda que realizada se encontró que los dos artículos que respondieron de mejor manera a esta pregunta fueron: “*Statistical and Machine Learning forecasting methods: Concerns and ways forward*” [2] y “*Short-Term Traffic Flow Forecasting: An Experimental Comparison of Time-Series Analysis and Supervised Learning*” [3]

El primer artículo que se estudiará [3] investiga el pronóstico de flujos de tráfico vehicular en el corto plazo, el cual es un área de investigación que ha tenido un gran desarrollo recientemente. En este ámbito se han publicado diversos trabajos e investigaciones que presentan diferentes enfoques de abordar el problema, los cuales por supuesto comparten ideas y características similares. En este artículo los autores abordan el problema en dos sentidos, el primero es revisar los enfoques existentes para las predicciones de flujo de tráfico al corto plazo bajo el análisis de modelos probabilísticos, o clásicos, y el segundo es la propuesta de los autores de modelos de regresión de máquinas de soporte vectorial (modelos de aprendizaje de máquinas).

Lo que se busca en la investigación es predecir las condiciones de tráfico de una red de transporte utilizando el comportamiento pasado (o los datos históricos), con lo cual, mejorar la calidad y precisión de estos pronósticos se convierte en un tema de importancia, no solo para informar a los pasajeros del transporte sobre las condiciones actuales y futuras sino para los agentes de logística puedan diseñar y realizar la infraestructura al servicio de la movilidad programada. Entonces formalmente se tienen que, para una secuencia dada de una cantidad física observada, como es la velocidad o el flujo de transporte en un nodo del transporte, el problema consiste predecir el valor de esta para un tiempo  $t$  futuro.

En este sentido el primer conjunto de técnicas que surgen para abordar este problema desde el enfoque de series de tiempo es la metodología propuesta por Box y Jenkins, el cual jugo un papel muy importante en el desarrollo de este área de investigación con sus enfoque ARMA (Media móvil autorregresiva, por sus siglas en ingles), los cuales derivaron en siguientes investigaciones y aproximaciones como los modelos de ARIMA (promedio móvil integrado autorregresivo, por sus siglas en ingles), modelos SARIMA, que son modelos ARIMA que incluyen la estacionalidad de la serie de tiempo, y otros modelos como el KARIMA, modelos GARCH ampliamente utilizados en pronósticos financieros, modelos VARMA y modelos STARIMA; estos dos últimos (VARIMA Y STARIMA) han sido propuestos para el pronóstico de tráfico de flujo con múltiples redes de nodos de transporte.

Por otro lado, para la solución de este problema también han sido utilizado modelos del mundo del aprendizaje de máquinas, como por ejemplo las redes neuronales artificiales, los modelos soporte de regresión vectorial (SVR), las redes bayesianas, regresiones no paramétricas y enfoques basados en algoritmos genéticos, los cuales son asociados a las redes neuronales por su enfoque en la optimización de la función de pérdida.

De esta manera, se presentó una revisión de los modelos que han sido utilizados para la solución del problema, el primero, desde el mundo clásico de las series de tiempo y sus diferentes variantes, y el segundo desde el mundo del aprendizaje de máquinas. En adelante los autores estudiaron un modelo propuesto utilizando máquinas de soporte de regresión vectorial (SVR) buscando modelar la estacionalidad de la serie al igual que el SARIMA. En este sentido los investigadores presentan dos nuevos modelos de regresión de vectores de soporte, diseñados específicamente para beneficiarse de la estacionalidad típica del flujo de tráfico y se muestra que presentan un trade-off interesante entre la precisión de la predicción y la eficiencia computacional

Los datos utilizados por los investigadores son experimentos tomados del Sistema de Medición del Desempeño de la Autopista de California, el cual tiene más de 30.000 sensores y detectores ubicados en nueve distritos de California. El sistema cuenta con 164 autopistas, 6328 estaciones detectoras de vehículos y 3470 detectores de rampa. La mayoría de los sensores son detectores únicos y generalmente miden el recuento de vehículos (flujo de tráfico) y la ocupación (cantidad de tiempo que el detector de circuito está activo).

Para el estudio de esta investigación se trabajó con nueve meses, desde enero de 2009 hasta septiembre del mismo año y con 16 estaciones seleccionadas al azar de todo el Distrito 7 de California. Los autores suavizaron la serie temporal agregando datos de flujo en intervalos de 15 minutos no superpuestos, lo que da como resultado 96 muestras por día por estación.

Para cada estación se utilizaron los primeros 4 meses, enero-abril de 2009, como período de entrenamiento, los dos meses siguientes, mayo-junio de 2009, como período de validación y ajuste de parámetros, y los tres meses finales, julio-septiembre, período de prueba. Solo 5 de estas 16 estaciones se usaron para realizar la selección del modelo. A continuación, se presenta por los autores:

Station	Highway	Lanes
715916	5 S	4
716312	91 E	6
716551	134 W	4
716841	605 N	4
<b>716933</b>	5 N	5
<b>717087</b>	10 W	4
<b>717123</b>	10 E	4
<b>717152</b>	10 E	4
<b>717190</b>	10 E	4
717269	60 W	5
718144	101 S	4
737344	10 E	4
763626	110 N	4
764151	5 S	4
767678	605 N	4
768682	5 N	5

*Conjunto De 16 Estaciones Utilizadas: estaciones utilizadas para la selección del modelo (se muestra en negrita)*

Es de anotar que solo se modelo con los días laborables en el conjunto de datos. Concluyen los autores que, dado que la distribución del flujo de tráfico durante las vacaciones es muy diferente con respecto a los días de trabajo, al incluir solo los días de trabajo, es posible comparar la estacionalidad diaria y semanal, por lo cual, de lo contrario, la estacionalidad diaria

habría sido menos significativa al comparar un día laboral con un día festivo.

Como se mencionó anteriormente, la meta consistía en predecir el flujo del tráfico 15 minutos antes de una marca de tiempo dada. Para esto, los investigadores utilizaron los siguientes modelos:

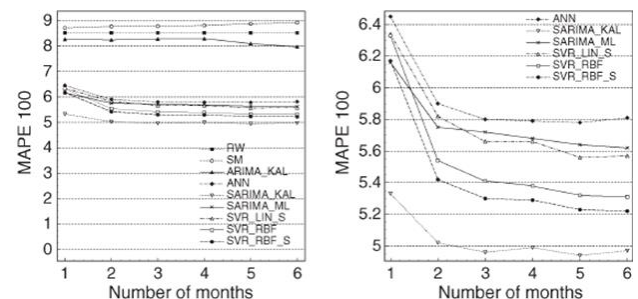
- RW: Es una línea de base que pronostica el tráfico en el futuro como igual a las condiciones actuales, matemáticamente como:  $X_t = X_{t-1}$
- SM: Modelo que predice para un momento determinado del día el promedio en el conjunto de entrenamiento.
- ARIMA: ARIMA con filtro Kalman, se nota como ARIMAKal por los investigadores.
- SARIMA: Un modelo ARIMA teniendo en cuenta estacionalidad con ajuste de máxima verosimilitud, se nota como SARIMAML por lo investigadores.
- SARIMA: SARIMA anterior con filtro Kalman (SARIMAKal).
- ANNs: Redes neuronales artificiales.
- SVR: Soporte vectoriales de regresión con núcleo RBF (SVRRBF).
- SVR con kernel RBF multiplicado por un kernel estacional (SVRS RBF).
- SVR con núcleo estacional lineal (SVRS lin).

Se realizó un primer experimento con el objetivo de comparar núcleos RBF simples con núcleos estacionales y, al mismo tiempo, probar la robustez de los parámetros del núcleo en múltiples estaciones del conjunto de datos empleado. En estos experimentos, los autores midieron MAPE100, donde este es el error porcentual absoluto medio, la formula se presenta a continuación:

$$MAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{X_t - \hat{X}_t}{X_t} \right|$$

Donde  $X_t$  es el flujo pronosticado. En MAPE100, la suma se calcula solo en los términos donde  $X_t > 100$  vehículos / hora, por lo tanto, se tiene en cuenta la medición de los errores donde aquellos períodos en los que el flujo de tráfico es significativo. Por otro lado, MAPExx se usa en la predicción de tráfico para evitar el problema de que, en pocos ejemplos con flujo de tráfico bajo, se podría afectar en gran medida la medición de los errores.

Ahora bien, se utiliza el conjunto de entrenamiento en los diferentes algoritmos, para lo cual se construye una curva de aprendizaje midiendo el rendimiento de cada clasificador en función del número de meses de entrenamiento. A continuación, se presenta la curva de estos rendimientos por cada algoritmo presentados por autores.



Curvas de aprendizaje de los algoritmos donde: el error de predicción (MAPE100) en función del número de meses utilizados en el conjunto de lluvia es presentado a la Izquierda y todos los algoritmos de mejor rendimiento se presentan a la derecha.

A continuación, se presenta los detalles sobre estos experimentos. SARIMAKal es el mejor modelo en términos de precisión; no obstante, el rendimiento esperado no mejora al incluir al conjunto de entrenamiento más de dos meses, por otro lado, SVRS RBF es el segundo mejor modelo y puede aprovechar un conjunto de entrenamiento más grande.

RW	SM	SARIMA <sub>ML</sub>	SARIMA <sub>Kal</sub>	ARIMA <sub>Kal</sub>	SVR <sub>RBF</sub>	SVR <sub>RBF</sub> <sup>S</sup>	SVR <sub>lin</sub> <sup>S</sup>	ANN
8.52(0.59)	8.71(0.93)	6.16(0.71)	<b>5.33</b> (0.68)	8.26(0.68)	6.33(0.75)	6.17(0.75)	6.34(0.69)	6.45(0.59)
8.52(0.58)	8.77(0.86)	5.75(0.59)	<b>5.02</b> (0.45)	8.24(0.80)	5.54(0.47)	5.42(0.49)	5.82(0.50)	5.90(0.48)
8.52(0.58)	8.78(0.86)	5.72(0.59)	<b>4.96</b> (0.46)	8.28(0.73)	5.41(0.47)	5.30(0.47)	5.66(0.49)	5.80(0.45)
8.52(0.58)	8.81(0.86)	5.68(0.60)	<b>4.99</b> (0.47)	8.29(0.72)	5.38(0.46)	5.29(0.47)	5.66(0.50)	5.79(0.43)
8.52(0.58)	8.88(0.87)	5.64(0.58)	<b>4.94</b> (0.47)	8.10(0.74)	5.32(0.48)	5.23(0.47)	5.56(0.47)	5.78(0.39)
8.52(0.58)	8.92(0.87)	5.62(0.59)	<b>4.97</b> (0.47)	7.97(0.48)	5.31(0.49)	5.22(0.48)	5.57(0.50)	5.81(0.35)

Curva de aprendizaje para todos los algoritmos comparados, usando 1 a 6 meses de entrenamiento y empleando los tres meses de prueba.

Station	RW	SM	SARIMA <sub>ML</sub>	SARIMA <sub>Kal</sub>	ARIMA <sub>Kal</sub>	SVR <sub>RBF</sub>	SVR <sub>RBF</sub> <sup>S</sup>	SVR <sub>lin</sub> <sup>S</sup>	ANN
715916	9.02	9.24	6.75	<b>5.71</b>	9.49	6.01	5.99	6.27	6.31
716312	8.22	8.55	5.66	<b>4.89</b>	8.15	5.21	5.19	5.80	5.99
716551	9.47	9.58	6.22	<b>5.35</b>	7.97	5.80	5.76	6.29	6.20
716841	9.04	8.31	5.58	<b>4.99</b>	9.35	5.46	5.36	5.74	5.85
716933	7.66	8.91	5.67	<b>5.02</b>	7.59	5.39	5.35	5.62	5.69
717087	8.33	7.83	5.38	<b>4.80</b>	8.27	5.21	5.20	5.44	5.51
717123	7.75	7.87	4.98	<b>4.50</b>	7.07	4.85	4.77	5.07	5.25
717152	8.51	8.66	5.80	<b>5.16</b>	8.18	5.62	5.54	5.87	6.00
717190	8.30	9.30	5.60	<b>5.08</b>	8.13	5.46	5.38	5.75	5.75
717269	8.97	11.02	6.67	<b>5.47</b>	9.09	6.01	5.81	6.24	6.24
718144	9.31	9.75	6.05	<b>5.37</b>	8.00	5.75	5.59	6.09	6.15
737344	8.81	9.16	6.20	<b>5.14</b>	8.88	5.86	5.75	5.86	6.35
763626	7.94	8.44	4.75	<b>4.21</b>	7.24	4.76	4.55	4.91	5.28
764151	8.34	8.04	5.69	<b>4.96</b>	7.93	5.28	5.11	5.46	5.79
767678	9.14	8.42	5.99	<b>4.94</b>	9.41	5.59	5.40	5.66	5.82
768682	7.58	7.46	4.56	<b>3.79</b>	7.73	4.22	4.12	4.45	4.59

Resultados detallados para las 16 estaciones de prueba, usando tres meses para el entrenamiento y tres meses para la prueba.



Se nota que SARIMAML funciona peor que SARIMAKal y SVR con un núcleo RBF. Por otro lado, las redes neuronales en este caso son menos precisas que las SVR simples. Los autores también analizan el rendimiento de cada algoritmo durante períodos que generalmente presentan altos niveles de congestión. Se elegirá entonces según los picos de la mañana entre las 7 a.m. y las 8 a.m. el cual midió el error de predicción para cada estación de prueba. Los valores que son altos de MAPE de los predictores RW y SM indican dificultad de en la meta de predicción. Es de notar que, en este caso, SARIMAKal no siempre predice de mejor manera en cada estación, ya que, en algunos lugares, los modelos de SVR obtiene un mejor rendimiento.

No obstante, se debe considerar el tiempo computacional al momento de construir un sistema de monitoreo de tráfico, que en este sentido y considerando el momento actual, es un tema a gran escala, considerando tanto el tiempo de entrenamiento como el tiempo de predicción. Por tal motivo, los autores consideran importante tener en cuenta este tiempo, el cual se presenta a continuación para todos los algoritmos considerando la configuración con tres meses de entrenamiento de los modelos y tres meses para prueba

Model	Training time (s)	Prediction time (s)	Error (MAPE <sub>100</sub> )
ARIMA <sub>Kal</sub>	6.8	0.4	8.3
SARIMA <sub>Kal</sub>	608.9	8.9	5.0
SARIMA <sub>ML</sub>	11.3	0.5	5.7
ANN	70.6	0.7	5.8
SVR <sub>Slin</sub>	18.7	0.1	5.7
SVR <sub>RBF</sub>	219.0	1.6	5.4
SVR <sub>RBF</sub>	276.2	10.7	5.3
SM	–	0.2	8.8
RW	–	0.1	8.5

*Tiempo de entrenamiento y pronóstico para todos los algoritmos, es un promedio de los 16 nodos considerados en el conjunto de prueba*

Se presenta entonces tanto el tiempo de predicción, que se tiene en cuenta en el momento de los pronósticos en tiempo real, como para el tiempo de entrenamiento, que puede convertirse en un cuello de botella si el objetivo es construir un modelo a gran escala.

Así entonces, el modelo SARIMAKal es que requiere el mayor tiempo de entrenamiento, se nota que es más del doble que para SVRS RBF, esto se debe al ajuste del modelo de espacio de estado de Kalman y el cálculo de la matriz de covarianza. Por otro lado, el modelo SARIMAML es más rápido. En este orden de ideas, el modelo SVRSlin, a pesar de lograr una precisión de pronóstico un poco mas más bajo, es más rápido que los modelos SARIMAKal y SVRRBF esto se debe al aprovechamiento del algoritmo de optimización de gradiente estocástico, por lo tanto, este podría representar una solución de compensación entre el coste computacional y la precisión de la predicción.

El segundo artículo [3] que se estudiará es en la evaluación de desempeño en múltiples horizontes de pronóstico utilizando un gran subconjunto de 1045 series de tiempo mensuales utilizadas en la competencia M3, el cual busca comparar la precisión de modelos clásicos y modelos de aprendizaje de Machine Learning. Los investigadores afirman que en el campo de la inteligencia artificial se han encontrado aplicaciones del pronóstico de series tiempo y se han realizado investigaciones sobre cómo se puede avanzado a partir de los resultados de investigaciones realizadas y sus conclusiones.

En el mundo del Aprendizaje Automático se ha encontrado literatura especialmente en el estudio de las redes neuronales artificiales con el fin de mejorar predicciones en series de tiempo.

Los modelos de Machine Learning al igual que los enfoques clásicos buscan mejorar la precisión del pronóstico al minimizar alguna función de pérdida, típicamente se utiliza la suma de los errores al cuadrado, la diferencia radica en cómo se realiza la minimización con los métodos de Machine Learning, los cuales utilizan algoritmos no lineales para esta tarea mientras que los modelos clásicos utilizan soluciones lineales.

A continuación, se presentan los métodos clásicos y de Machine Learning estudiados por los autores para abordar este problema. En primer lugar, se incluye seis métodos de la competencia M3, los más precisos encontrados por los investigadores, más un punto de referencia ingenuo. En este sentido, el modelo Naïve 2 es uno de ellos, que un modelo de caminata aleatoria ajustado por estacionalidad. El suavizado exponencial simple, es el segundo, el cual realiza predicción sin tener en cuenta la estacionalidad de la serie. Como tercer y cuarto modelo se aborda el suavizado exponencial Holt y Damped, que son modelos apropiados para series de tiempo teniendo en cuenta el componente estacione. En el quinto lugar se tiene una combinación de tres métodos de suavizado exponencial: SES, Holt y Damped buscando posibles beneficios de promediar los errores de pronósticos múltiples. Además, para un sexto modelo se utiliza el método Theta el cual logra una mejora en general en sMAPE. Por último, los algoritmos ARIMA y suavizado exponencial obtienen el séptimo y octavo puesto, los cuales, según los investigadores, han demostrado tener una precisión considerable.

Ahora bien, se presentan los modelos de Aprendizaje Automático empleados en este estudio para realizar la comparación de precisión en los pronósticos. En primer lugar, se presenta el modelo de Perceptrón Multicapa (Redes neuronales), luego se emplea una Red Neuronal Bayesiana, en tercer lugar se utiliza funciones de base radial, que es una red de avance con una capa oculta, en cuarto lugar se utilizan redes neuronales de regresión generalizada, posteriormente, un quinto modelo consiste en K vecinos más cercano o KNN, luego se emplean métodos de árboles de regresión CART, más adelante se modelo de soporte de regresión vectorial son empleados o también llamados SVR, por últimos los autores utilizan modelos de procesos gaussianos, redes neuronales recurrentes y redes neuronales de memoria a corto y largo plazo.

Para el procesamiento de los datos, los autores emplean las siguientes consideraciones:

- Datos originales: Los investigadores no aplican ningún procesamiento previo.
- Transformación de los datos: Se calcula la potencia de Box-Cox a datos originales con el fin de lograr estacionariedad en la varianza.
- Desestacionalizar los datos: Los datos se desestacionalizan utilizando el enfoque clásico de descomposición multiplicativa.
- Reducción de tendencia de datos: Los autores prueba de Cox-Stuart para establecer si se debe emplear tendencia lineal

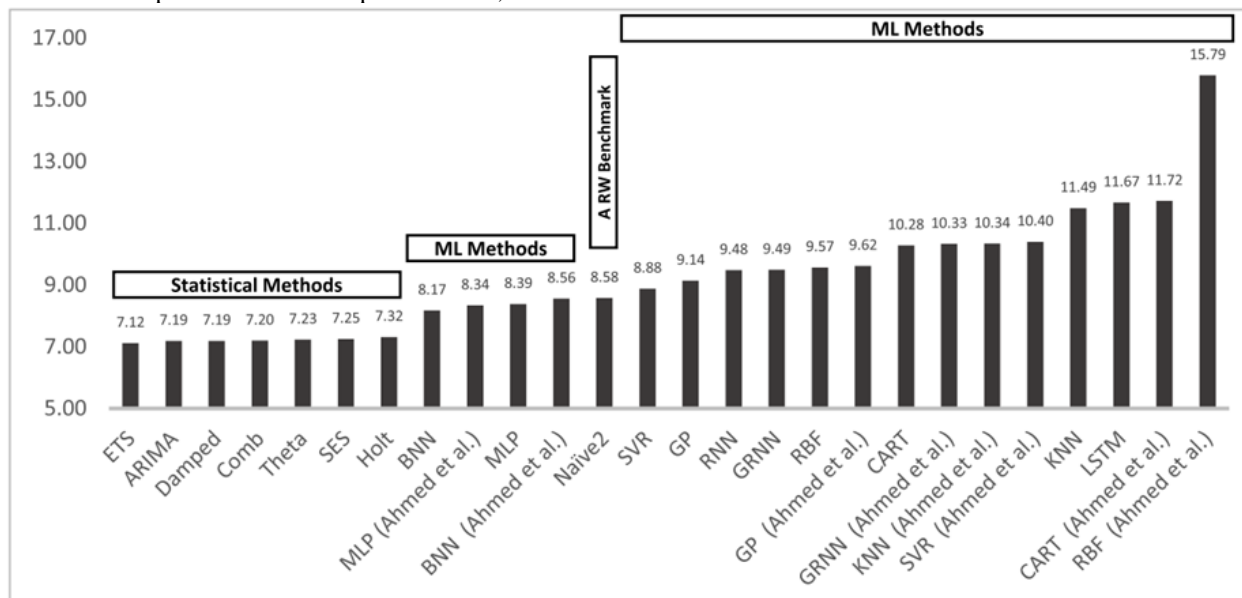
determinista, o en otro caso la primera diferenciación, con el fin de eliminar tendencia de los datos y lograr la estacionariedad en la media.

- Combinación de proceso: Aplica, según los investigadores, utilizar los tres anteriores pasos con el fin de buscar beneficios en las técnicas de preprocesamiento individuales.

A partir de estas consideraciones los autores realizan lo que denominamos un pre modelamiento, el cual incluyen probar rápidamente distintos modelos, para buscar que transformaciones pudieran ser más representativas, en sentido

de la calidad del pronóstico, a la hora de seleccionar las técnicas adecuadas del preprocesamiento de los datos que sea adecuada, y nos permita ser comparable para los enfoques del estudio, esto se realiza tanto para los modelos propuestos de ML como para los modelos de enfoque clásico.

A continuación, se presenta los resultados para la medición del SMAPE:



Se observa que los métodos más precisos son los clásicos estadísticos, por encima de los propuestos de ML para este estudio, los autores afirman que los algoritmos de aprendizaje han revolucionado una amplia gama de aplicaciones en diversos campos y no hay razón para que no se pueda lograr lo mismo con los métodos de ML en el pronóstico. Por lo tanto, la comunidad investigación debe enfocar sus esfuerzos a como encontrar la aplicación de estos modelos para mejorar su capacidad de predicción con una mayor precisión.

En este sentido, los autores afirman que, si hubo alguna forma de aprendizaje, la precisión de los métodos de ML debería haber excedido la de ARIMA y superar ampliamente a la Naive2. Se hace énfasis sobre cómo se puede estudiar y a su vez explicar que los métodos de aprendizaje automático puedan aprender del comportamiento futuro en lugar de sobre ajustar los datos pasados.

## B. Resultados pregunta 2: ¿Qué técnicas de Machine Learning han sido usadas con éxito en series de tiempo?

En el análisis de la primera pregunta se encontró que los métodos clásicos como ETS y ARIMA superan los métodos aprendizaje automático en pronósticos de series tiempo. Sin embargo, también se observó que hay casos en los que las técnicas de Machine Learning ofrecen ventajas sobre los métodos tradicionales, particularmente en pronósticos que requieren múltiples periodos futuros (multi-step forecasting). Ahora bien, el propósito de la segunda pregunta consiste en indagar por los mejores algoritmos de Machine Learning en los que se haya probado su buen desempeño en predicciones de series temporales.

De la búsqueda realizada los artículos que mejor respondieron a esta pregunta fueron: “An Empirical Comparison of Machine Learning Models for Time Series Forecasting” [4] y “Machine Learning Strategies for Time Series Forecasting” [5]. Así mismo, en los artículos de la pregunta 1, específicamente en el artículo “Statistical and Machine Learning forecasting methods: Concerns and ways forward” [3], también se presentan métodos de Machine Learning y su desempeño probado en series de tiempo, por lo tanto, también aportarán a responder a esta pregunta.

En el artículo [4], los autores realizan un estudio de comparación de varias técnicas de Machine Learning aplicadas a los datos mensuales de la Competencia M3, una competencia organizada por el IFF (International Institute of Forecasters, ver <https://forecasters.org/>), en la cual se presentan 1045 series de tiempo de distintas índoles como industria, finanzas, demografía y macroeconomía. Es interesante mencionar que estos son los mismos datos usados en el artículo [3] que retornó la búsqueda de la primera pregunta. En este artículo los investigadores compararon los siguientes algoritmos de aprendizaje: Perceptrón Multicapa (MLP), Redes Neuronales Bayesianas (BNN), Funciones de Base Radial (RBF), Red Neuronal de Regresión Generalizada Funcional (GRNN), K-Vecinos más Cercanos (KNN), Árboles de Clasificación y Regresión (CART), Regresión de Soporte Vectorial (SVR) y Procesos Gaussianos (GP).

Es importante mencionar el preprocesamiento y la ingeniería de atributos a la que fueron sometidos los datos antes de entrenar los modelos. A nivel de preprocesamiento los investigadores usaron 3 técnicas:

- Transformación logarítmica de los datos.

- *Desestacionalización.* Midieron la autocorrelación con un retraso (lag) de 12 meses, y probaron la hipótesis de "no estacionalidad" con la fórmula de Bartlett. Si la prueba indicaba estacionalidad, entonces aplicaban una descomposición aditiva clásica descrita en el artículo [6].
- *Normalización.* Mediante un escalamiento lineal de los datos entre -1 y 1.

Posteriormente usaron 3 métodos para transformar las series. Los resultados finales se evaluaron para cada una de estas transformaciones:

- *LAGGED-VAL:* los valores de las series de tiempo se retrasan un periodo y el valor a predecir (salida objetivo) es el siguiente valor.
- *DIFF:* calcular la diferencia de cada valor con el del periodo anterior.
- *MOV-AVG:* promedios móviles con ventanas de suavizado de diferentes tamaños.

Finalmente se debe mencionar que la métrica de evaluación seleccionada fue el *Error Porcentual Absoluto Medio Simétrico* (SMAPE por sus siglas en inglés), calculado mediante la siguiente expresión:

$$SMAPE = \frac{1}{M} \sum_{m=1}^M \frac{|\hat{y}_m - y_m|}{(|\hat{y}_m| + |y_m|)/2}$$

donde  $y_m$  es la variable de salida y  $\hat{y}_m$  es la predicción. Esta métrica fue usada en un procedimiento de evaluación k-fold de 10 validaciones. La métrica promedio entre las 10 validaciones fue llamada *Overall SMAPE* y la métrica promedio entre las 1045 series fue llamada *SMAPE-TOT*.

A continuación, se presentan los resultados de los diferentes algoritmos para cada una de las transformaciones que realizaron sobre las series, cabe resaltar que la métrica SMAPE mide un error y por lo tanto indica un mejor desempeño del modelo mientras menor sea su valor:

Rendimiento general en todas las series temporales 1045 para el método de preprocesamiento LAGGED-VAL:

Model	SMAPE-TOT	Mean Rank	Rank Interval	FRAC-BEST
MLP	0.0857	2.78	(2.62,2.94)	35.6
BNN	0.1027	3.42	(3.26,3.58)	16.9
RBF	0.1245	5.33	(5.17,5.49)	6.7
GRNN	0.1041	5.24	(5.08,5.40)	6.3
KNN	0.1035	5.10	(4.94,5.26)	7.3
CART	0.1205	6.77	(6.61,6.93)	3.2
SVR	0.0996	4.19	(4.03,4.35)	8.3
GP	0.0947	3.17	(3.01,3.33)	15.7

Rendimiento general en todas las series temporales 1045 para el método de preprocesamiento DIFF:

Model	SMAPE-TOT	Mean Rank	Rank Interval	FRAC-BEST
MLP	0.1788	5.02	(4.86,5.18)	6.5
BNN	0.1749	4.58	(4.42,4.74)	14.7
RBF	0.2129	5.24	(5.08,5.40)	8.8
GRNN	0.1577	3.92	(3.76,4.08)	14.7
KNN	0.1685	4.57	(4.41,4.73)	11.7
CART	0.1529	3.93	(3.77,4.09)	26.1
SVR	0.1709	4.59	(4.43,4.75)	6.7
GP	0.1654	4.15	(3.99,4.31)	10.7

Rendimiento general en todas las series temporales 1045 para el método de preprocesamiento MOV-AVG:

Model	SMAPE-TOT	Mean Rank	Rank Interval	FRAC-BEST
MLP	0.0834	2.88	(2.72,3.04)	35.4
BNN	0.0858	2.94	(2.78,3.10)	15.9
RBF	0.1579	6.28	(6.12,6.44)	4.9
GRNN	0.1033	4.99	(4.83,5.15)	4.7
KNN	0.1034	4.95	(4.79,5.11)	8.4
CART	0.1172	6.31	(6.15,6.47)	3.8
SVR	0.1040	4.41	(4.25,4.57)	6.7
GP	0.0962	3.26	(3.09,3.42)	20.0

Se observa que el preprocesamiento DIFF conduce a peores resultados que los otros dos métodos. Los modelos MLP (Perceptrón Multicapa) y GP (Procesos Gaussianos) son los dos mejores modelos, siendo MLP el mejor para ambos métodos de preprocesamiento LAGGED-VAL y MOV-AVG y GP siendo más robusto porque también funcionó muy bien para el preprocesamiento DIFF.

Con la investigación de este archivo se encontraron 2 algoritmos de Machine Learning con un buen desempeño para pronósticos de un solo paso en series de tiempo. Ahora bien, el segundo artículo encontrado, "*Machine Learning Strategies for Time Series Forecasting*" [5], trata el problema de pronósticos de múltiples pasos.

Los autores de este artículo se centraron en investigar los pronósticos mediante técnicas de Aprendizaje Local, los cuales "intentan ajustar localmente la capacidad del sistema de entrenamiento a las propiedades del conjunto de entrenamiento en cada área del espacio de entrada. La familia de estos algoritmos contiene métodos conocidos, como el método de k-vecinos más cercanos (kNN) o las redes de función de base radial (RBF)" [7]. Esta decisión tuvo los siguientes motivos:

- *Número reducido de supuestos:* el aprendizaje local supone que no hay conocimiento a priori sobre el proceso subyacente a los datos. Por ejemplo, no asume la existencia de una función global que describe los datos y no se asume ninguna de las propiedades del ruido.
- *Capacidad de aprendizaje en línea:* el método de aprendizaje local puede manejar fácilmente las tareas de aprendizaje en línea donde el número de muestras de capacitación aumenta con el tiempo.

Este artículo [5], a diferencia del anterior, no pretende poner a prueba diferentes algoritmos para compararlos, sino que reúne, después de una revisión de literatura, los mejores métodos para abordar para el problema de la predicción multi-step con métodos de Machine Learning. Los autores proponen estos 4 métodos:

- Estrategia Recursiva:* entrena primero un modelo one-step y luego lo usa de forma recursiva para devolver una predicción de varios pasos.
- Estrategia Directa:* aprende independientemente  $H$  modelos y devuelve una predicción multi-step concatenando las  $H$  predicciones.
- Estrategia DirRec:* combina las 2 estrategias anteriores. Esta calcula los pronósticos con diferentes modelos para cada horizonte (como la estrategia Directa) y, en cada paso de tiempo, amplía el conjunto de entradas agregando variables correspondientes a los pronósticos del paso anterior (como la estrategia Recursiva).

- iv. *Estrategias de múltiple salida*: el valor pronosticado ya no es una cantidad escalar sino un vector de valores futuros de la serie temporal. Principalmente:
  - a. *Estrategia MIMO*. La estrategia Multi-Input Multi-Output (MIMO, también conocida como estrategia conjunta) evita la suposición simplista de la independencia condicional entre los valores futuros realizada por la estrategia directa al aprender un único modelo de salida múltiple.
  - b. *Estrategia DIRMO*. Tiene como objetivo preservar los aspectos más atractivos de las estrategias DirRec y MIMO al articular el horizonte H en varios bloques y al usar MIMO para pronosticar los valores dentro de cada bloque.

## CONCLUSIÓN

Habitualmente se suelen considerar al Aprendizaje Automático y al Aprendizaje Profundo como la solución clave para todos los problemas de modelado predictivo. En el caso particular de las series temporales, aunque los métodos de Machine Learning son capaces de realizar predicciones con un desempeño aceptable, la evidencia muestra que los métodos clásicos superan a los métodos de Aprendizaje Automático y Profundo [2]. Esto es consistente con una de las conclusiones de la competencia M3, que establece que los modelos simples tienden a superar a los modelos más complejos en la predicción de series temporales [4].

Si bien varios estudios coinciden en el anterior hallazgo, también se encontró que los métodos tradicionales tienen 2 limitaciones: en primer lugar, la mayoría de los métodos clásicos resuelven el problema de predicción de un solo paso, lo cual hace que estos pronósticos se queden cortos en modelos para sistemas de alerta temprana. En segundo lugar, en algunos casos, hay correlaciones entre variables en datos de series de tiempo multivariadas, y a menudo se obtiene una mejor comprensión al modelar todas las variables relacionadas juntas que al modelar una sola variable [1].

Respondiendo a las preguntas de investigación se puede concluir que los Métodos Clásicos de pronóstico de series temporales ofrecen un mejor desempeño que los métodos de Machine Learning sobre series de tiempo univariantes y de pronóstico de un solo paso. Pero cuando se requieren pronósticos de pasos múltiples o pronósticos sobre series multivariadas, los métodos de Machine Learning ofrecen fortalezas para superar las limitaciones que los métodos clásicos encuentran estos problemas. Sin embargo, se

encontró que la literatura sobre el caso particular de las series de tiempo multivariadas todavía es limitada y se considera que la investigación futura debería estar relacionada con los pronósticos de series multivariadas.

Finalmente, se ha encontrado en la literatura de pronósticos de modelos de Machine Learning que la mayoría de los estudios publicados proporcionan los pronósticos sin compararlos con métodos estadísticos clásico. Hacerlo aumenta las expectativas de que los métodos de Machine Learning brinden predicciones precisas, pero sin ninguna prueba empírica de que este sea el caso. Además de las pruebas empíricas, es necesario trabajo de investigación con el fin de comprender cómo se generan los pronósticos de los métodos de ML, que es el mismo caso con los modelos de inteligencia artificial donde se tienen métricas muy precisas, pero en numerosas ocasiones no es posible explicar cómo se obtiene esta salida. De esta manera, no es aceptable trabajar con modelos de “cajas negras” ya que es necesario saber cómo surgen los pronósticos y cómo pueden ser influenciados o ajustados para llegar a predicciones viables.

## REFERENCIAS

- [1] Du, S., Li, T., Yang, Y., & Horng, S. J. (2020). Multivariate Time Series Forecasting via Attention-based Encoder-Decoder Framework. *Neurocomputing*.
- [2] Lippi, M., Bertini, M., & Frasconi, P. (2013). Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning. *IEEE Transactions on Intelligent Transportation Systems*, 14(2), 871-882.
- [3] Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and Machine Learning forecasting methods: Concerns and ways forward. *PloS one*, 13(3).
- [4] Ahmed, N. K., Atiya, A. F., Gayar, N. E., & El-Shishiny, H. (2010). An empirical comparison of machine learning models for time series forecasting. *Econometric Reviews*, 29(5-6), 594-621.
- [5] Bontempi, G., Taieb, S. B., & Le Borgne, Y. A. (2012, July). Machine learning strategies for time series forecasting. In *European business intelligence summer school* (pp. 62-77). Springer, Berlin, Heidelberg.
- [6] Makridakis, S., Wheelwright, S. C., & Hyndman, R. J. (2008). *Forecasting methods and applications*. John Wiley & sons.
- [7] Bottou, L., & Vapnik, V. (1992). Local learning algorithms. *Neural computation*, 4(6), 888-900.