**Pronóstico de Series de Tiempo de Tráfico web. Estudio de Caso Universidad Nacional de Misiones**

POR: Enrique Gauto Sand

Propuesta de Tesis presentada a la Facultad de Ciencias Exactas, Químicas y Naturales de la Universidad Nacional de Misiones para optar al grado académico / título profesional de Licenciado en Sistemas de Información de la carrera de Licenciatura en Sistemas de Información

Directora: Alice Rambo

Prof. Tit. de la Cátedra “Trabajo Final”: Dr. Kuna, Horacio Daniel

Índice

[Capítulo 1 1](#_Toc159172867)

[Introducción 1](#_Toc159172868)

[1.1. Motivación 2](#_Toc159172869)

[1.2. Objetivos 2](#_Toc159172870)

[1.2.1. Objetivo General: 2](#_Toc159172871)

[1.2.2. Objetivos Específicos: 2](#_Toc159172872)

[1.3. Estructura del documento 2](#_Toc159172873)

[Capítulo 2 3](#_Toc159172874)

[Marco Teórico 3](#_Toc159172875)

[2.1. Tráfico web 3](#_Toc159172876)

[2.2. Series de tiempo 3](#_Toc159172877)

[2.1. Pronóstico de series de tiempo de tráfico web 3](#_Toc159172878)

[2.2. Evaluación de desempeño de los modelos predictivos 5](#_Toc159172879)

[Capítulo 3 6](#_Toc159172880)

[Descripción del problema 6](#_Toc159172881)

[3.1 El problema 7](#_Toc159172882)

[3.2 Estructura Organizacional 7](#_Toc159172883)

[3.3 Organigrama (checar) 7](#_Toc159172884)

[Capítulo 4 9](#_Toc159172885)

[Solución Propuesta 9](#_Toc159172886)

[4.1 Materiales y Métodos 10](#_Toc159172887)

[4.2 Herramientas y Materiales 11](#_Toc159172888)

[4.2.1 Python 11](#_Toc159172889)

[4.2.2 Google Drive 11](#_Toc159172890)

[4.2.3 Universal Analytics (UA) 11](#_Toc159172891)

[4.2.4 Google Analytics 4 (GA4) 11](#_Toc159172892)

[4.2.5 Looker Studio 11](#_Toc159172893)

[4.2.6 Word 11](#_Toc159172894)

[4.2.7 Excel 11](#_Toc159172895)

[4.2.8 Numpy 12](#_Toc159172896)

[4.2.9 Tensorflow 12](#_Toc159172897)

[4.2.10 Keras 12](#_Toc159172898)

[4.2.11 Pandas 12](#_Toc159172899)

[4.2.12 Matplotlib 12](#_Toc159172900)

[4.2.13 Google Colaboratory 12](#_Toc159172901)

[4.2.14 Git 12](#_Toc159172902)

[4.2.15 Hyperparametros 13](#_Toc159172903)

[4.2.16 Epoch 13](#_Toc159172904)

[4.2.17 Trial 13](#_Toc159172905)

[4.2.18 GRU 13](#_Toc159172906)

[4.2.19 Github 13](#_Toc159172907)

[4.3 Comprensión de los datos 14](#_Toc159172908)

[4.3.1 Recolección de datos iniciales 14](#_Toc159172909)

[4.3.2 Preparación de los datos 15](#_Toc159172910)

[4.2 Modelos 1](#_Toc159172911)

[3.5 Selección de Modelos 1](#_Toc159172912)

[Bibliografía 4](#_Toc159172913)

# Capítulo 1

# Introducción

## Motivación

## Objetivos

### Objetivo General:

Pronosticar series de tiempo de tráfico web y medir el desempeño de modelos aplicados sobre datos extraídos de las páginas institucionales de la U.Na.M.

### Objetivos Específicos:

-Relevar y analizar distintos modelos de pronóstico de series de tiempo de tráfico web.

-Determinar la adecuación de cada uno de los modelos propuestos de la bibliografía a la problemática.

-Definir indicadores para medir el desempeño de los modelos.

-Generar los modelos a utilizar.

-Realizar la evaluación del desempeño de los modelos.

-Validar los modelos.

## Estructura del documento

# Capítulo 2

# Marco Teórico

En este capítulo se presenta el marco teórico del trabajo realizado, comenzando con la definición de tráfico web y series de tiempo, luego con variados ejemplos de la literatura de distintas tecnologías que realizan el pronóstico de series de tiempo de tráfico web finalizando con las formas de medir el desempeño de los modelos.

## Tráfico web

El trafico web es generado por los usuarios de una página web, el trafico web son los datos que se envían y se reciben correspondientes a los usuarios que visitan la página web[1].

## Series de tiempo

Las series de tiempo son un conjunto de valores medidos en orden secuencial en el tiempo[2]; cuando se miden los datos para generar una serie de tiempo, generalmente se toman los valores con la misma separación en tiempo entre cada valor[3].

## Pronóstico de series de tiempo de tráfico web

El pronóstico de series de tiempo se basa en las observaciones pasadas de la serie de tiempo a pronosticar y otras entradas, siendo el proceso de predecir valores futuros de una serie de tiempo [3].

Entre las formas del pronóstico de series de tiempo de tráfico web los autores en[4] proponen de redes *Generative Adversarial model* (GAN)[4] con *Long Short Term Memory* (LSTM)[5] y un perceptrón multicapa (MLP)[6], donde LSTM y GAN actuarían como generador y el MLP como discriminador, para generar series de tiempo dado el conjunto de datos real, finalmente se realizaría el pronóstico con la librería *Prophet*[7] comparando la combinación de tecnologías anteriores con métodos estadísticos, finalmente llegando a la conclusión de que los autores no obtuvieron una diferencia notable[4].

Entre las formas del pronóstico de series de tiempo de tráfico web existe proponen[8] la técnica de redes neuronales *Long Short Term Memory* (LSTM)[5] con entrenamiento asíncrono distribuido, cuya métrica de desempeño que se utilizó según el autor en[8] es la MAE (*mean absolute error*)[9] y la función de pérdida de Huber[10] para probar la precisión del modelo, han logrado un buen grado de asertividad con 200 épocas, en el documento científico[8] menciona obtuvieron una MAE en promedio menor que 30, lo que consideran un buen resultado[8].

Hay otro artículo[1] en el cual se menciona el uso de (*Recurrent Neural Networks*) RNN seq2seq[1] con la ayuda de la arquitectura *encoder/decoder*, el *encoder* es cuDNN GRU (*Gate Recurrent Unit*)[11] ya que realiza la tarea con mayor velocidad en comparación con los tensores regulares, el *decoder* es TensorFlow GRUBlockCell[11], en este documento científico[1] menciona que realizaron algunos cambios al modelo ganador de *Kaggle*[11] como agregar la mediana de 7,30,90 y 180 días, usaron para medir el desempeño del modelo la métrica *Symmetric Mean Absolute Percentage Error* (SMAPE)[12] donde comentan los autores que obtuvieron un SMAPE de 0.349[1].

En otro artículo[13], proponen utilizar un enfoque híbrido de *particle swarm optimization* y *quantum-inspired* resultando en QPSO[13] y redes *Nonlinear autoregressive exogenous* NARX[14] resultando en NARX -QPSO[13] en dicho artículo se menciona que el modelo NARX –QPSO tiene un rendimiento sobresaliente en los resultados comparados con otros modelos[13].

En otro artículo[15] propone un diseño de un sistema *Automated Machine Learning* (Auto ML) [15], una arquitectura neuronal nueva denominada Auto-PyTorch-TS[15] donde lo comparan con otros modelos y demuestran que su modelo tiene un mejor rendimiento[15].

También se propone en el artículo[16] el uso de un modelo combinado

*Prophet* y *light gradient boosting machine* (LGBM)[16] donde se menciona que tiene mejores resultados comparado con modelos individuales[16].

Otra tecnología son las redes bidireccionales LSTM (BI-LSTM)[2] las cuales en este artículo[2] fueron probadas en el conjunto de datos *M3-Competition*[17]el cual se usa para probar modelos de pronóstico de series de tiempo dado que posee distintas categorías, en el artículo[2] mencionan que probaron las redes BI-LSTM sobre el conjunto de datos M3 en el periodo trimestral comparándolo con el modelo *Auto-Regressive Integrated Moving Average* (ARIMA)[3] y otros, donde BI-LSTM tuvo un mejor rendimiento.

## Evaluación de desempeño de los modelos predictivos

El desempeño es la forma de evaluar el modelo, se utiliza para comparar los valores predichos del modelo entrenado, con los valores observados[18]

El *Root Mean Square Errors* (RMSE): el error cuadrático medio es una medida de las diferencias entre los valores predichos por un modelo y los valores observados.

Donde es el valor predicho y es el valor real[9]

El MAE (*mean absolute error*)[9] es una métrica que sirve para medir el error medio absoluto.

Donde es el valor predicho y es el valor real[9],

La métrica *Symmetric Mean Absolute Percentage Error* (SMAPE)[12] básicamente es el error absoluto medio porcentual simétrico.

Siendo el valor pronosticado, es el valor observado y n es el tamaño de la muestra.

# Capítulo 3

# Descripción del problema

En este capítulo se describe la problemática a abarcar como así también la estructura organizacional.

## **El problema**

El presente trabajo pretende hacer análisis de tráfico web para poder estimar y predecir futuras demandas en las páginas institucionales de la Universidad Nacional de Misiones – U.Na.M. Para lo cual se consideraran indicadores como volumen de tráfico, cantidad de conexiones, cantidad de visitantes, velocidad de carga de las paginas, páginas vistas, promedio de páginas por vista, porcentaje de rebote, entre otros indicadores según la disponibilidad de los mismos. Siendo provisto por parte del personal técnico referente institucional acceso a las herramientas de captura de datos estadístico de los siguientes enlaces:

* Sitio Institucional de la U.Na.M. disponible en <https://unam.edu.ar/>
* Sitio de la editorial Universitaria disponible en <https://editorial.unam.edu.ar/>
* Portal de acceso al contenido generado por la radio y la televisión de la Universidad disponible en <https://transmedia.unam.edu.ar/>

De los sitios mencionados anteriormente se extraerán los datos para utilizar en este proyecto con la finalidad de realizar el pronóstico de series de tiempo de tráfico web. Inicialmente como herramienta de acceso a los datos se dispone *google analytics* con permisos cedidos por los administradores.

## Estructura Organizacional

Con el fin de e1 entender el funcionamiento del Departamento de Gestión de Recursos de Redes y Comunicaciones se procede a describir su estructura organizacional, como así también los roles de los trabajadores.

## Organigrama (checar)



Figura 1: Organigrama del Departamento de Gestión de Recursos de Redes y Comunicaciones

El Departamento de Gestión de Recursos de Redes y Comunicaciones, gestiona la infraestructura, además pone a disposición las herramientas para la puesta en servicio de las páginas web y servicios añadidos. Pero no controla cómo los contenidos, diseño o función de los sitios.

* **Jefe de departamento**: Realiza las tareas propias del jefe de departamento, también se encarga de mantener las plataformas de los sitios web, por ejemplo Moodle para Aulas Virtuales.
* **Técnico 1:** Se encarga principalmente de la gestión y mantenimiento de los servicios de VoIP, redes WiFi, gestión de las VLAN y redes de la unidad central de Rectorado, gestión de usuarios en la plataforma de Google Workspace for education, entre otros.
* **Técnico 2 y Técnico 3:** Estos dos técnicos se encargan de la gestión  y ruteo de las redes de la universidad y con los proveedores de servicio, gestión y mantenimiento de los Servidores físicos y virtuales del CPD, plataformas de Backup, sistemas de monitoreo de las plataformas y servicios esenciales de la infraestructura de la universidad como DNS, BGP. OSPF, VPN, etc.

# Capítulo 4

# Solución Propuesta

En este capítulo se describe la metodología seleccionada para la solución, como así también las herramientas utilizadas, y además los modelos.…

## Materiales y Métodos

Para el desarrollo del presente trabajo se utilizará la metodología CRISP-DM[19] como guía, ya que hoy en día es un estándar de facto para los proyectos de ciencia de datos y minado de datos[20].

|  |  |  |
| --- | --- | --- |
| **Actividad arreglar** | **Tiempo Estimado** | **Referencia** |
| Relevar y analizar distintos modelos de pronóstico de series de tiempo de tráfico web.   * Selección de los buscadores académicos * Generación de filtros de búsqueda * Aplicación de filtros de búsqueda y evaluación de resultados. | 2 mes | A |
| Tratamiento de los datos | 6 meses | B |
| Determinar la adecuación de cada uno de los modelos propuestos de la bibliografía a la problemática.   * Listar los modelos propuestos resultantes de la búsqueda bibliográfica. * Comparar los modelos. * Seleccionar los modelos que se adecuen a la problemática. | 2 mes | C |
| Definir indicadores para medir el desempeño de los modelos.   * Realizar una búsqueda bibliográfica de los indicadores para medir el desempeño. * Analizar y seleccionar los indicadores para medir el desempeño. | 1 mes | D |
| Generar los Modelos   * Generar la prueba de diseño. * Construir los modelos. | 2 meses | E |
| Realizar la evaluación del desempeño de los modelos.   * Evaluar y ajustar los modelos. | 4 mes | F |
| Validar los modelos. | 2 mes | G |
| Realizar la escritura del documento de tesis. | 22 meses | H |
| TOTAL | 22 meses | |

## Herramientas y Materiales

### Python

Python según dice en su página oficial es un lenguaje de programación interpretado, soporta múltiples paradigmas como pudiera ser programación funcional o programación orientada a objetos, Python es un lenguaje multipropósito, es decir, sirve para solucionar distintos tipos de problemas[21], para este trabajo se utilizaron las versiones de Python 3.7.x y 3.10.x.

### Google Drive

Google ofrece una plataforma que permite almacenar archivos para uso personal a sus usuarios en la nube, este servicio además permite compartir archivos entre usuarios de Google Drive, como así también la edición de archivos en tiempo real con las aplicaciones que nos permite conectar[22].

### Universal Analytics (UA)

Esta es una propiedad de Analytics, básicamente según el soporte de Google cuando se registran vistas u otro tipo de datos del trafico web se recopilan en la propiedad; Universal Analytics es la generación anterior de Analytics desde el 1 Julio del 2023 dejó de procesar datos[23].

### Google Analytics 4 (GA4)

GA4 es una propiedad de Analytics, según el soporte de Google, GA4 recoge datos y los procesa de las páginas web y aplicaciones, mientras que UA no va a funcionar más GA4 es la nueva generación, GA4 por su parte recompila datos desde antes de la fecha en la que UA dejo de funcionar[24].

### Looker Studio

Según la documentación de Google es una herramienta para realizar informes y visualizar los datos, esta herramienta permite la conexión de fuente de datos propios de los usuarios como pudiera ser Google Analytics[24], también se pueden exportar datos de tablas de informes de Looker Studio en archivos CSV(Autoría propia).

### Word

### Excel

### Numpy

Según su documentación oficial es una librería *open source,* es un estándar para trabajar con datos numéricos en Python, la librería posee estructuras de datos como Arrays multidimensionales, Matrices, además posee funciones para realizar cálculos matemáticos con dichos Arrays y matrices[25].

### Tensorflow

Es una biblioteca según su documentación oficial es de código abierto para el aprendizaje automático[26].

### Keras

Es una biblioteca que sirve de API  (Aplication Programing Interface) de alto nivel de Tensorflow, actualmente keras se encuentra dentro de la biblioteca Tensorflow y es una api de alto nivel oficial de Tensorflow[26].

### Pandas

Es una librería de Python para el análisis y manipulación de datos, posee el objeto dataframe, también funciones para manipular datos del dataframe y funciones para leer y escribir archivos CSV (valores separados por coma)[27].

### Matplotlib

Matplotlib según su página oficial es una librería para la creación de gráficas, animaciones, distintas formas de visualización de datos en Python[28].

### Google Colaboratory

Según su página oficial Colaboratory es un producto de Google Research que permite a cualquier usuario escribir y ejecutar código de Python en la nube. Este producto se pude conectar con google drive para leer los archivos de google drive[29].

### Git

Git es un sistema de control de versiones, sirve para controlar las versiones de un proyecto, un proyecto es un repositorio de Git, un proyecto comprende todo el conjunto de carpetas y archivos al que se le realiza el control de versiones, de manera que se guardan los cambios y también se pueden recuperar las versiones anteriores del proyecto[30].

### Hyperparametros

Son las variables que se utilizaron durante el entrenamiento del modelo, como así también pueden definir la topología del modelo, estas variables tienen un impacto en el rendimiento del modelo[31].

### Epoch

Según la documentación oficial de keras un Epoch, o traducido el español como época, en este documento de ahora en más se le llamará época, es cuando se pasa por todo el conjunto de datos[32], es decir, una época es cuando el modelo hace una pasada por todo el conjunto de datos, en keras, se puede agregar callbacks al final de una época para guardar o imprimir información, como por ejemplo guardar el modelo, de forma que en un futuro dado x cantidad de épocas, se pueda recuperar el modelo con mejor rendimiento.

### Trial

Según su documentación oficial, un trial es una prueba, donde cada trial tiene x cantidad de épocas, en un trial se evalúa un conjunto de valores de hyperparámetros, es decir, en cada trial se evalúa una configuración diferente del modelo[33].

### GRU

### Github

Github según su página oficial es una plataforma que hospeda repositorios de Git, ofreciendo además que los equipos de desarrollo puedan trabajar en grupo[30].

## 4.3 Comprensión de los datos

Esta fase inicia con recolectar los datos y continúa con actividades para familiarizarse con los datos, identificar problemas de calidad de los datos, descubrir los primeros conocimientos de los datos[34].

### 4.3.1 Recolección de datos iniciales

Se tienen acceso a datos históricos de las siguientes 3 páginas webs:

* Sitio Institucional de la U.Na.M. disponible en <https://unam.edu.ar/>
* Sitio de la editorial Universitaria disponible en <https://editorial.unam.edu.ar/>
* Portal de acceso al contenido generado por la radio y la televisión de la Universidad disponible en <https://transmedia.unam.edu.ar/>

De las cuales tenemos mayor cantidad de datos de registrados por UA y menor cantidad de datos expresados en días registrados en GA4, las siguientes tablas explican los datos correspondientes por fecha que se tienen de dichas páginas web.

|  |  |  |
| --- | --- | --- |
| Página Web | Desde | Hasta |
| <https://unam.edu.ar/> | 28-05-2018 | 04-07-2023 |
| <https://editorial.unam.edu.ar/> | 08-10-2018 | 04-10-2023 |
| <https://transmedia.unam.edu.ar/> | 22-08-2018 | 01-06-2023 |

Tabla de datos de Universal Analytics

|  |  |  |
| --- | --- | --- |
| Página Web | Desde | Hasta |
| <https://unam.edu.ar/> | 14-06-2022 | 23-11-2023 |
| <https://editorial.unam.edu.ar/> | 14-06-2022 | 24-01-2024 |
| <https://transmedia.unam.edu.ar/> | 14-06-2022 | 01-01-2024 |

Tabla de datos de Google Analytics 4

Para poder descargar los datos se procedió a conectar Google Analytics como fuente de datos de Looker Studio, de esta manera se podían pasar los datos a tablas en Looker Studio y posteriormente descargar los datos en archivos CSV.

### 4.3.2 Preparación de los datos

Identificando algunos problemas de calidad de los datos la página web <https://unam.edu.ar/> desde noviembre del año pasado no está capturando el tráfico correctamente en GA4, por lo que se procede a descartar este conjunto de datos.

Lo siguiente que se procedió a realizar, comenzando por la página web <https://transmedia.unam.edu.ar/> se descargaron los datos en archivos csv del Looker Studio de UA, son 5 archivos CSV que se procede a unificar en un solo archivo a través de código en Python.

Se descargaron los siguientes datos en 5 archivos CSV:

* Vistas por Sistemas Operativos
* Vistas por Agrupación de canales predeterminados
* número de vistas de página
* Usuarios Nuevos
* Usuarios
* Numero de Sesiones Por usuario
* Sesiones
* Vistas por Categoría de dispositivo
* Vistas por País

Se procedieron a transformar los datos obtenidos de los archivos CSV de Looker Studio para obtener las columnas del dataframe.

De vistas por sistema operativo salieron las siguientes columnas:

* Windows
* Android
* Linux
* Macintosh
* Sistema operativo (not set)
* iOS
* Windows Phone
* Chrome OS
* Tizen
* Sistema operativo promedio
* Sistema operativo std

Siendo Sistema operativo (not set) donde se registran las vistas los sistemas operativos que UA no pudo detectar, Sistema operativo promedio el promedio de las otras columnas de sistemas operativos, Sistema operativo std el desvio estándar del promedio y el resto son vistas de sistemas operativos conocidos, por ejemplo la columna Windows tendrá en cada fila datos de las vistas de los usuarios que accedieron a la página web a través de un dispositivo cuyo sistema operativo es Windows.

De vistas por Agrupación de canales predeterminados salieron las siguientes columnas:

* Organic Search
* Social
* Direct
* Referral
* Agrupación de canales predeterminada promedio
* Agrupación de canales predeterminada std

Donde Organic Search, Social, Direct y Referral son los canales predeterminados, es decir, las fuentes de tráficos más comunes[35], Agrupación de canales predeterminada promedio es el promedio de los canales predeterminados, y Agrupación de canales predeterminada std es el desvío estándar.

De vistas por Categoría de dispositivo salieron las siguientes columnas:

* Mobile
* Desktop
* Tablet
* Categoría de dispositivo promedio
* Categoría de dispositivo std

Donde Mobile, Desktop, Tablet son la Categoría de dispositivo, es decir, el tipo de dispositivo que tenía el usuario cuando se registró la visita en la página web[36], además Categoría de dispositivo promedio es el promedio de vistas y Categoría de dispositivo std es el desvió estándar.

Las siguientes son tal cual del CSV no se necesitó mayor trabajo de transformación:

* número de vistas de página
* Usuarios Nuevos
* Usuarios
* Numero de Sesiones Por usuario
* Sesiones

Es decir una columna de Usuarios Nuevos del CSV se pasó al dataframe como una columna de Usuarios Nuevos sin mayor preprocesado.

De vistas por País se procedió a dividir en 2 columnas, las vistas que son de Argentina y las que son de otros países.

Luego, se procedió a confirmar la existencia de NaN (Not an number), y eliminar las filas que posean NaNs,El siguiente paso en el preprocesado fue agregar la columna mes, la cual número del mes, luego las columnas de correlación anual que es el promedio de vistas por año y trimestral que básicamente son el promedio de vistas por trimestre.

Finalmente se procedió a calcular el predictor Flag (no se como citar esto)

El predictor flag se calcula tomando el min y máximo de los datos, de ese rango lo divido por un número, en este caso se comenzó con 30 y se pasó a 15, esta división me da el rango de vistas por etiqueta, la etiqueta del rango será el valor medio, luego se crea una columna llamada predictor Flag, en la que por cada fila de la columna de total de vistas (las vistas reales medidas por Google Analytics) se le asigna una etiqueta que coincida con un rango previamente calculado.

Donde RPE es el rango por etiqueta y N es el número de etiquetas, las etiquetas que se utilizaron posteriormente es el valor medio de cada rango; cuando una vista coincide dentro de un rango, se le asigna la etiqueta correspondiente a dicho rango.

Quedando como resultado del preprocesado las siguientes columnas de tum transmedia, que fueron las mismas columnas que se tomaron en cuenta para las otras páginas web de la institución.

Listado de columnas del Dataframe:

* Predictor Flag
* Número de vistas de página
* Usuarios nuevos
* Usuarios
* Número de sesiones por usuario
* Sesiones
* Argentina
* Otros Paises
* Mobile
* Desktop
* Tablet
* Categoría de dispositivo promedio
* Categoría de dispositivo std
* Windows
* Android
* Linux
* Macintosh
* Sistema operativo (not set)
* iOS
* Windows Phone
* Chrome OS
* Tizen
* Sistema operativo promedio
* Sistema operativo std
* Organic Search
* Social
* Direct
* Referral
* Agrupación de canales predeterminada promedio
* Agrupación de canales predeterminada std
* Mes
* correlacion anual
* correlacion Trimestral

Siendo un total de 33 columnas, por supuesto que desde Google Analytics se registran más datos, sin embargo debido al cambio de UA a GA4[37] solo se eligieron los datos que a nivel tanto conceptual como de medición, fueran más parecidos entre sí.

Finalmente se realizó una limpieza de los outliers detectados usando KNN.(describir en detalle o lo mando a anexo?)

## **4.2 Modelos**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tecnologías** | **Red Neuronal** | **Resultados  según Artículos científicos** | **Finalidad** | **Aplica** |
| *Multi-Layer Perceptron*  (MLP) | SI | - | Clasificación, predicción | No |
| *Long-Short Term Memory*  (LSTM) | SI | Combinándolo con otras tecnologías se tiene mejor resultado que usarlo individualmente. | Predictivo, hallar patrones | No |
| ARIMA | No | Buenos | No es redes neuronales, es estadística  predictiva (predicción de series temporales) | No |
| *Gate recurrent unit* (GRU) | SI | El modelo ganador de Kaggle utiliza un cuDNN GRU que es más rápido que un GRU normal. | Sistemas predictivos basados en información secuencial. | No |
| *Bidirectional-LSTM* | SI | Mejor rendimiento comparándolo en el *M3-Competition* | Reconocimiento y clasificación | No |
| GAN | SI | Regular, no hay ganancia comparado con ARIMA | Creación artificial | No |
| RNN seq2seq | SI | Buenos | Modelo ganador del concurso de *kaggle* de *web traffic* 2017 | Si |
| NARX -QPSO | SI | Buenos | Predecir el siguiente valor de la señal de entrada | Si |
| *Prophet-LGBM* | SI | Buenos Mejores resultados que modelos individuales | Combinación ligthGBM y Prophet para predecir trafico web | Si |

Tabla 1 Comparación de tecnologías

## **3.5 Selección de Modelos**

En la Tabla 1 se presenta una comparación de las tecnologías, de ahí se procede a seleccionar las tecnologías que si aplican, son las que se consideran que se podrían usar para la solución del problema; la razón de descartar algunas tecnologías son porque no usan redes neuronales, otras son tecnologías cuya finalidad no es la del pronóstico de series de tiempo y por último, de acuerdo a los artículos científicos presentados en el marco teórico, algunas tecnologías no tienen buen rendimiento o el rendimiento suficiente para ser tomado en cuenta.

Entre los modelos que quedan seleccionados están RNN seq2seq, NARX –QPSO y Prophet-LGBM de estos modelos Prophet-LGBM si bien se mencionan las fechas de los datos que utilizaron no está disponible el dataset para compararlo con los datos de la problemática de esta tesis por lo tanto Prophet-LGBM se descarta como posible tecnología para la solución de la problemática de esta tesis, mientras que por otro lado tanto RNN seq2seq y NARX –QPSO ambos utilizan el mismo conjunto de datos el cual está disponible online para que cualquiera lo pueda descargar, de esta manera y dado que RNN seq2seq es el ganador del concurso de Kaggle(citar) y por lo tanto se encuentra más información de la misma en la página de Kaggle se opta por utilizar esta tecnología.

En cuanto a la métrica seleccionada para la medición del desempeño de los modelos, elijo la métrica SMAPE, debido a que es la misma con la que se evaluó el modelo RNN seq2seq, que es el ganador del concurso de Kaggle.

**Plan de pruebas**

El plan de prueba consiste generar varios modelos GRU con la arquitectura seq2seq, y se procederá a medir usando la métrica SMAPE, en primera instancia se utilizarán los datos de tum transmedia de UA, posteriormente se utilizará el mejor modelo resultante en los datos de tum transmedia de GA4, también en los datos de editorial universitaria UA y el mejor modelo resultante de editorial universitaria UA se utilizará para generar un modelo con los datos de editorial universitaria GA4.

(nose si utilizar la palabra reentrenar, generar, entrenar, transferir, entre otros)

**Construcción de los Modelos**

En esta sección se detallan las características de los modelos generados, básicamente en primera instancia se generaron 4 modelos por cada algoritmo de Hypertuner, siendo los algortimos de Hypertuner Hyperband, Bayesian optimisation y el de Random Search. Siendo un total de 12 modelos, tomando datos de 7, 14, 21 y 28 días anteriores para pronosticar los siguientes 7 días, además, se partieron los datos 80% para entrenamiento y 20% para prueba.

**Cantidad de Trials por Modelo**

Cada trial tiene 100 epochs y se corrieron un total de 50 trials por modelo. Respecto a la medición, por cada epoch se tiene un rendimiento, entonces por cada trial solo se guarda la epoch con mejor rendimiento.

**Hyperparametros que se Afinaron**

|  |  |  |
| --- | --- | --- |
| Nombre | Valores posibles | Definición |
| units | 128-512 con un paso de 32 | Es la cantidad de unidades de GRU por capa. |
| lr | Entre 1e-4 y 1e-2 | Es el ratio de aprendizaje, *Learning Rate* en ingles  Se utiliza un Optimizador Adam |
| dropout | 0-0.5 con un paso de 0.1 | El dropout apaga cierto número de neuronas, para evitar el sobre entrenamiento |
| recurrent\_dropout | 0-0.5 con un paso de 0.1 | Apaga un número aleatorio de conexiones recurrentes, para evitar el sobrenentrenamiento |
| batch\_size | 32 a 128 con un paso de 32 | Tamaño de lote a tomar, normalmente, esto es mara dividir la cantidad de muestras a tomar, ya que no se puede pasar todo el conjunto de datos entero de una sola vez. |

**Evaluación**

# Bibliografía

[1] N. Petluri and E. Al-Masri, “Web Traffic Prediction of Wikipedia Pages,” *Proc. - 2018 IEEE Int. Conf. Big Data, Big Data 2018*, pp. 5427–5429, 2019, doi: 10.1109/BigData.2018.8622207.

[2] D. Quoc Nguyen, M. Nguyet Phan, and I. Zelinka, “Periodic Time Series Forecasting with Bidirectional Long Short-Term Memory: Periodic Time Series Forecasting with Bidirectional LSTM,” *ACM Int. Conf. Proceeding Ser.*, pp. 60–64, 2021, doi: 10.1145/3453800.3453812.

[3] V. Kotu and B. Deshpande, “Chapter 12 - Time Series Forecasting,” V. Kotu and B. B. T.-D. S. (Second E. Deshpande, Eds. Morgan Kaufmann, 2019, pp. 395–445.

[4] K. Zhou, W. Wang, L. Huang, and B. Liu, “Comparative study on the time series forecasting of web traffic based on statistical model and Generative Adversarial model,” *Knowledge-Based Syst.*, vol. 213, p. 106467, Feb. 2021, doi: 10.1016/J.KNOSYS.2020.106467.

[5] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/NECO.1997.9.8.1735.

[6] K. Zhou, W. Wang, L. Huang, and B. Liu, “Comparative study on the time series forecasting of web traffic based on statistical model and Generative Adversarial model,” *Knowledge-Based Syst.*, vol. 213, Feb. 2021, doi: 10.1016/J.KNOSYS.2020.106467.

[7] “Prophet: forecasting at scale.” https://research.facebook.com/blog/2017/2/prophet-forecasting-at-scale/ (accessed Oct. 24, 2020).

[8] R. Casado-Vara, A. M. del Rey, D. Pérez-Palau, L. De-La-fuente-valentín, and J. M. Corchado, “Web Traffic Time Series Forecasting Using LSTM Neural Networks with Distributed Asynchronous Training,” *Math. 2021, Vol. 9, Page 421*, vol. 9, no. 4, p. 421, Feb. 2021, doi: 10.3390/MATH9040421.

[9] W. Wang and Y. Lu, “Analysis of the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE) in Assessing Rounding Model,” in *IOP Conference Series: Materials Science and Engineering*, 2018, vol. 324, no. 1, doi: 10.1088/1757-899X/324/1/012049.

[10] G. P. Meyer, “An Alternative Probabilistic Interpretation of the Huber Loss,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5257–5265, doi: 10.1109/CVPR46437.2021.00522.

[11] “Web Traffic Time Series Forecasting,” 2017. https://www.kaggle.com/c/web-traffic-time-series-forecasting/discussion/39367 (accessed Oct. 12, 2022).

[12] “Choosing the correct error metric: MAPE vs. sMAPE,” 2020. https://towardsdatascience.com/choosing-the-correct-error-metric-mape-vs-smape-5328dec53fac (accessed Oct. 12, 2022).

[13] C. Kuranga and N. Pillay, “A comparative study of nonlinear regression and autoregressive techniques in hybrid with particle swarm optimization for time-series forecasting,” *Expert Syst. Appl.*, vol. 190, Mar. 2022, doi: 10.1016/j.eswa.2021.116163.

[14] P. Khanarsa, A. Luangsodsai, K. Sinapiromsaran, I. F. Astachova, K. A. Makoviy, and Y. V Khitskova, “Possibilities for predicting the state of usability web resources,” *J. Phys. Conf. Ser.*, vol. 1902, no. 1, p. 012029, May 2021, doi: 10.1088/1742-6596/1902/1/012029.

[15] D. Deng, F. Karl, F. Hutter, B. Bischl, and M. Lindauer, “Efficient Automated Deep Learning for Time Series Forecasting,” 2022, Accessed: Oct. 04, 2022. [Online]. Available: https://github.com/automl/Auto-PyTorch.

[16] S. Xu, C. Han, and C. Ran, “A Time Series Combined Forecasting Model Based on Prophet-LGBM,” *ACM Int. Conf. Proceeding Ser.*, May 2021, doi: 10.1145/3469213.3470280.

[17] “The M3-Competition Database.” https://forecasters.org/resources/time-series-data/m3-competition/ (accessed Oct. 18, 2022).

[18] A. Botchkarev, “Performance Metrics (Error Measures) in Machine Learning Regression, Forecasting and Prognostics: Properties and Typology,” 2018, [Online]. Available: http://arxiv.org/abs/1809.03006.

[19] P. Chapman *et al.*, “CRISP-DM 1.0: Step-by-step data mining guide,” *SPSS inc*, vol. 9, no. 13, pp. 1–73, 2000.

[20] C. Schröer, F. Kruse, and J. M. Gómez, “A systematic literature review on applying CRISP-DM process model,” *Procedia Comput. Sci.*, vol. 181, pp. 526–534, 2021, doi: 10.1016/J.PROCS.2021.01.199.

[21] Python Software Foundation, “Python 3.12.1 documentation.” https://docs.python.org/3/ (accessed Feb. 02, 2024).

[22] Google, “Plataforma de archivos compartidos y almacenamiento personal en la nube - Google.” https://www.google.com/intl/es-419\_ar/drive/ (accessed Feb. 02, 2024).

[23] Google, “Ayuda de Google.” https://support.google.com/ (accessed Feb. 02, 2024).

[24] Google Lcc, “Ayuda de Google,” *Ayuda de Google*, 2023. https://support.google.com/.

[25] Numpy.org, “NumPy Documentation.” https://numpy.org/doc/ (accessed Feb. 05, 2024).

[26] Google Lcc, “TensorFlow es una plataforma de código abierto de extremo a extremo para el aprendizaje automático.” https://www.tensorflow.org/ (accessed Feb. 05, 2024).

[27] “About pandas.” https://pandas.pydata.org/about/ (accessed Feb. 06, 2024).

[28] “https://matplotlib.org/.” https://matplotlib.org/ (accessed Feb. 06, 2024).

[29] Google, “Google Colab.” https://research.google.com/colaboratory (accessed Feb. 02, 2024).

[30] GitHub, “Acerca de Git.” https://docs.github.com/es/get-started/using-git/about-git (accessed Feb. 06, 2024).

[31] “Introduction to the Keras Tuner.” https://www.tensorflow.org/tutorials/keras/keras\_tuner (accessed Feb. 15, 2024).

[32] “Keras FAQ.” https://keras.io/getting\_started/faq/# (accessed Feb. 15, 2024).

[33] “The base Tuner class.” https://keras.io/api/keras\_tuner/tuners/base\_tuner/ (accessed Feb. 15, 2024).

[34] W. Rüdiger and J. Hipp, “CRISP-DM : Towards a Standard Process Model for Data Mining,” *Proc. Fourth Int. Conf. Pract. Appl. Knowl. Discov. Data Min.*, no. 24959, pp. 29–39, 2000.

[35] “[UA] Información sobre las agrupaciones de canales.” https://support.google.com/analytics/answer/6010097 (accessed Feb. 07, 2024).

[36] “[GA4] Informe ‘Detalles de la tecnología.’” https://support.google.com/analytics/answer/12980150 (accessed Feb. 07, 2023).

[37] “[UA→GA4] Diferencias entre los datos de Universal Analytics y Google Analytics 4.” https://support.google.com/analytics/answer/9964640 (accessed Feb. 10, 2024).