Universidad EAFIT

Maestría en Ciencia de Datos y Analítica

Aprendizaje Automático

S2261-0136

Aprendizaje Supervisado Modelos Híbridos

Autores:

Alejandro Barrientos-Osorio

Luis Miguel Caicedo-Jimenez

Omar Alejandro Henao-Zapata

21 de abril de 2022

I. INTRODUCCIÓN

La predicción efectiva de valores de consumo energético juega un papel muy importante en la distribución de recursos efectiva para suplir con el consumo, reducir desabastecimiento y optimizar costos [1]. Este documento pretende comparar varios modelos de regresión supervisados para entender cual modelo predice mejor el fenómeno y también, para verificar si los modelos híbridos basados en separar partes lineales y no lineales del modelo influyen en una mejora significativa del poder predictivo general.

Aprendizaje supervisado es un rama del machine learning (ML) que se enfoca en una función que mapea un input a un output basado en ejemplos conocidos de pares input-output. [2] Esta también es llamada una tarea de inferencia (Contrario a descriptivo) de una función con unos datos de entrenamiento.

Separar los datos de entrenamiento, deja por fuera una fracción de los datos, llamados datos de testeo o prueba, sobre los cuales se prueba la función para evitar el sobre ajuste del problema y su generalización.

En nuestro caso particular, tenemos un base de datos de carga eléctrica en Bélgica, seleccionado debido a que en varios artículos mencionan que los modelos híbridos son una solución valida a problemas con componentes lineales y no lineales, como se observan en series de datos de carga eléctrica. La base de datos es extraída de `https://www.elia.be/en/grid-data`

A. Objetivo general

- Estudio de modelos híbridos y su comparación en desempeño en series de tiempo con otros modelos de Deep Learning (DL) o tradicionales.

B. Objetivos Específicos

- Comparación de modelos para la componente lineal entre auto-arima y Prophet desarrollado por Facebook.
- Realizar una ingeniería de características adecuada para series de tiempo.
- Comparación de modelos para la componente no lineal entre LSTM univariado y multivariado utilizando variables arrojados por la ingeniería de características.

- Implementación de una Backpropagation Neural Network (BPNN) como modelo híbrido para la predicción a partir de los resultados del modelo lineal y no lineal.

A partir de las respuestas a los objetivos, pretendemos concluir cual modelo se adapto mejor al problema. Todo el desarrollo fue realizado en Python notebooks encontrados en la carpeta de Google Drive y enviados junto con este reporte.

El resto del documento pretende explicar la metodología usada, un breve análisis de los resultados y finalmente concluir.

II. METODOLOGÍA

La metodología que se usó fue CRISP-DM, conocida para afrontar problemas de desarrollo de analítica, sin embargo, el paso de implementación no fue llevado a cabo debido a que el propósito del trabajo es una comparación y estudio.

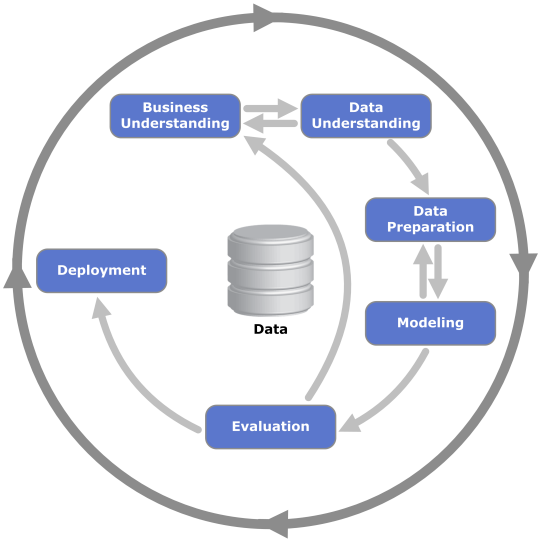


Figura 1: Metodología CRISP-DM

En la figura 1 se puede ver una representación gráfica de la metodología usada.

A. Entendimiento del Negocio

La predicción de carga es dividida generalmente en tres tipos: corto plazo (horas a semanas), mediano plazo (semanas a años), largo plazo (Más de años). Predicción a corto plazo

ha atraído bastante atención debido a que le facilita a las empresas prestadoras de servicios y distribuidoras llevar a cabo operaciones más optimizadas y seguras en la operación diaria del sistema. [1]

Es por esto, que la predicción se basó en una serie de datos con frecuencia por horas y ventanas cortas de tiempo.

B. Entendimiento de los datos

Los datos originalmente están con una frecuencia de muestreo cada 15 minutos. Se trata de una serie de tiempo univariada de tres columnas, fecha, carga eléctrica en MW en ese momento especifico y una columna adicional llamada Resolution code usada para identificar el área, que en nuestro caso es igual siempre, debido a que estamos estudiando solo la zona de influencia del distribuidor Elia . Este es un vistazo de las primeras 5 filas de los datos.

Datetime	Resolution code	Elia Grid Load
April 17, 2022 3:30 AM	PT15M	6,095.149 MW
April 17, 2022 3:45 AM	PT15M	6,121.895 MW
April 17, 2022 4:00 AM	PT15M	5,986.511 MW
April 17, 2022 4:15 AM	PT15M	6,218.309 MW
April 17, 2022 4:30 AM	PT15M	6,104.821 MW

Cuadro I: 5 primeras filas de datos extraídos de
`https://opendata.elia.be/explore/dataset/ods003/information/`

La medición de estos datos, está disponible desde el 31 de diciembre de 2014 hasta casi tiempo real, hay un retraso de alrededor de 5 horas.

Como nuestros son datos son de carga eléctrica, como explica el autor [1] tiene una parte lineal muy bien predicha por modelos simples y lineales, que tienen desventajas a la hora de predecir no linealidad y tienen la parte no lineal de la carga, que ha sido modelada por modelos basados en lógica difusa, maquinas de soporte vectorial o redes neuronales recurrentes (RNN), que tienen desventajas como alta complejidad, problemas de convergencia y mala predicción de la parte lineal, es por esto que se proponen los modelos híbridos.

C. Preparación de los datos

La preparación de datos se basó en algunas transformaciones, revisión y limpieza de los datos y finalmente en la ingeniería de características.

D. Transformaciones

Primero, realizamos una inversión de los datos, debido a que la fuente original estaba invertida, es decir, los registros más nuevos estaban primero, debido a la forma en la que se alimenta la información.

Luego, convertimos la columna Datetime a un formato tipo datetime en Python, ya que estaba siendo leído como una secuencia de caracteres. Sin embargo, los datos originales tenían también un componente de zona de tiempo, en este caso era UTC+0 o GMT+2. Decidimos trabajar todo en UTC debido a que no tenía alteraciones y eliminamos el parámetro timezone.

El paso siguiente fue realizar un corte de los datos desde el 1/1/2019 00:00 AM hasta el momento de extracción de los datos que fue el 26/2/2022 10:00 AM.

Finalmente, hicimos un reindexado de los datos para tener los datos cada hora (En vez de cada 15 minutos) por problemas de computación en el momento de entrenar las redes neuronales y del análisis multivariado, una breve explicación del problema de dimensionalidad es que en datos de carga eléctrica normalmente se trata con retrasos o lags de 24 horas (Esto en datos cada 15 minutos serían retrasos de 96 registros) con 8 características para cada ventana esto sería extraer ventanas de 768 para cada entrenamiento, teniendo en cuenta que entre 2019 y la fecha de extracción habrían 110636 registros. Esto es explicado con más profundidad en el modelado.

Ahora, específicamente para el entrenamiento y predicción de las redes neuronales recurrentes (RNN), los datos pasaron por una transformación por medio de un herramiento de escalado, en nuestro caso utilizamos siempre el MinMaxScaler del paquete de preprocesamiento de la librería scikit-learn.

E. Limpieza y Revisión

En este paso, ejecutamos dos pasos, revisión de datos duplicados y revisión de datos faltantes o NaN.

Datetime	Resolution code	Elia Grid Load
January 1, 2018 0:00 AM	PT15M	7,740.775 MW
January 1, 2018 0:00 AM	PT15M	7,740.775 MW
November 15, 2021 12:15 PM	PT15M	NaN
October 6, 2021 0:45 AM	PT15M	NaN

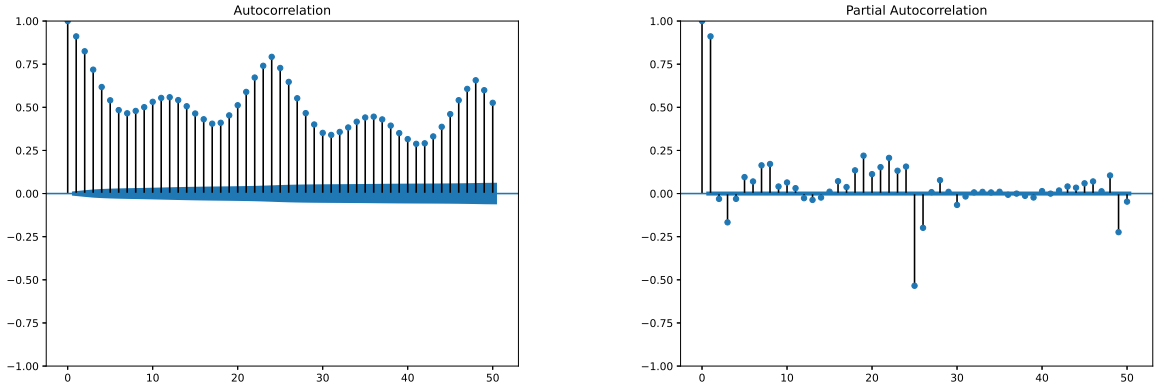
Cuadro II: Datos duplicados o con faltantes de la extracción original

Los datos duplicados fueron eliminados con el corte de datos desde 2019 en adelante y los datos faltantes, debido a que se hizo una extracción cada hora, también fueron eliminados. Teniendo en cuenta que los datos originales contaban con 250896 registros, consideramos que los datos son bastante confiables y que tienen la calidad suficiente como para continuar con la ingeniería de características.

F. Ingeniería de características

Este paso es usado para los modelos LSTM explicados en la parte de modelado, no se usaron para la predicción de modelos lineales debido a que son modelos poco robustos y valga la redundancia, lineales, y las componentes temporales adicionadas todas son altamente no lineales. Debido a que los modelos de RNN fueron usadas para la predicción de residuales en el modelo híbrido, toda la sección de ingeniería de características será mostrada con los resultados de los residuales. Esto será explicado en la parte de modelado. Además, en la parte de validación se aplicaron los mismos pasos para los datos originales, la validación se explicará con más detalle en secciones posteriores.

Como se mencionó anteriormente, la primera característica añadida, fue la de los retrasos debido a que en consumos eléctricos que miden patrones de consumo agrupado, hay mucha correlación de manera intuitiva cada 24 horas debido a que llega la noche y esto incrementa la carga sobre el sistema y distribuidores, esta hipótesis fue validada por medio de la figura 2(b).



((a)) Total ((b)) Parcial

Figura 2: Gráficas de autocorrelación total y parcial.

Aquí se puede tomar todos los retrasos desde el 1 hasta el 24, debido a que todos mostraron significancia estadística o al más representativo que sería el 24. En nuestro caso, para reducir dimensionalidad, escogimos incluir en los modelos solo el retraso 24, con resultados positivos, un estudio propuesto a futuro, sería incluir todos los retrasos del 1 al 24 y estudiar si hay una mejora en la precisión del modelo.

A continuación, realizamos un estudio de componentes temporales. Las RNN son excelentes al predecir secuencias, pero la estructura de datos temporales es contraintuitiva en estos términos, debido a que por ejemplo, en semanas del año que están definidas como el intervalo $(1, 52)$ y cogiendo como ejemplo los extremos, se podría ver como que la semana 1 es la más alejada de la semana 52, sin embargo, están solo a una semana de distancia. Esto mismo ocurre con los días del mes, las semanas del mes, el día del año, etc.

Es por esto, que utilizamos representaciones cosinusoidales para la inclusión de estos parámetros. Una representación gráfica de lo anteriormente explicado se puede ver en la figura 3. Los parámetros temporales incluidos para las regresiones, en especifico fueron:

- Hora
- Día del año
- Día del mes
- Mes del año

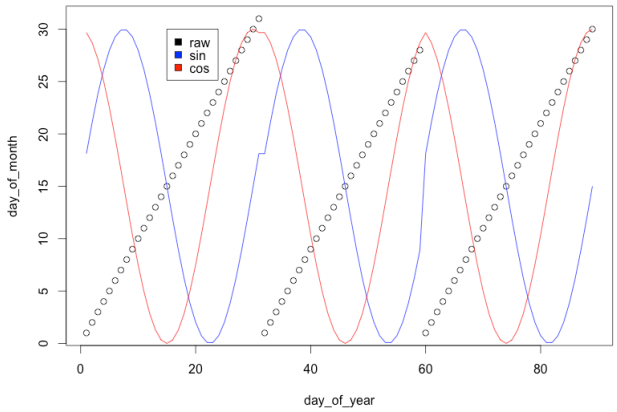


Figura 3: Gráfica de dispersión de día del mes en el eje Y vs día del año en el eje X

- Semana del año

Finalmente, incluimos una variable binaria codificada como 1 cuando fuera un día considerado como festivo y 0 cuando no. Las fechas de las festividades o eventos considerados significativos en Bélgica fueron extraídos del paquete Holidays en Python y adicionadas como una columna adicional al dataset.

G. Modelado

Se utilizaron varios modelos para la predicción de cada componente de los datos. Todas las comparaciones para la decisión se hicieron basados en MAPE (Mean absolute percentage error)

1. Componente lineal

En esta parte se compararon dos modelos, uno tradicional como lo es ARIMA, por medio del paquete pmdarima utilizamos la funcionalidad de auto-arima, el cual permitió encontrar por medio de criterios de información como AIC y BIC los mejores parámetros p, d, q para el modelo final.

El segundo usado fue Prophet desarrollado por Facebook que es un modelo bastante robusto y automatizado en el que solo hicimos una búsqueda de parámetro τ que según [3] indica escala de cambios de tendencia, es decir, cada cuanto puede cambiar la estimación de

la tendencia del modelo, esto es debido a que Prophet es un modelo lineal que divide la serie de tiempo en tres componentes: Tendencia, Estacionalidad, Festividades y un error normal aleatorio. [3].

Luego, de tener las predicciones realizadas de cada modelo, se escoge el modelo con más bajo MAPE y se calculan los residuales, que pasan como input a la parte no lineal del procedimiento.

2. Componente no lineal

Con los residuales del modelo escogido para la predicción lineal, realizamos dos predicciones por medio de LSTM. Los parametros del modelo univariado fueron:

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 24, 50)	10400
lstm (LSTM)	(None, 24, 50)	20200
lstm (LSTM)	(None, 24, 50)	20200
dense (Dense)	(None, 1)	51

Cuadro III: Total params and trainable params: 50,851. Resumen del modelo univariado RNN.

Para el LSTM multivariado, como se explicó en la ingeniería de características, se utilizarán 7 columnas con parámetros adicionales a la variable de carga. El resumen del modelo, se ve así:

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 24, 192)	154368
lstm (LSTM)	(None, 24, 192)	295680
lstm (LSTM)	(None, 192)	295680
dense (Dense)	(None, 1)	193

Cuadro IV: Total params and trainable params: 745,921. Resumen del modelo multivariado RNN.

Nuevamente, se comparan los MAPE de los dos modelos y se escoge el de métrica más baja para continuar como input a la parte final del procedimiento.

3. BPNN

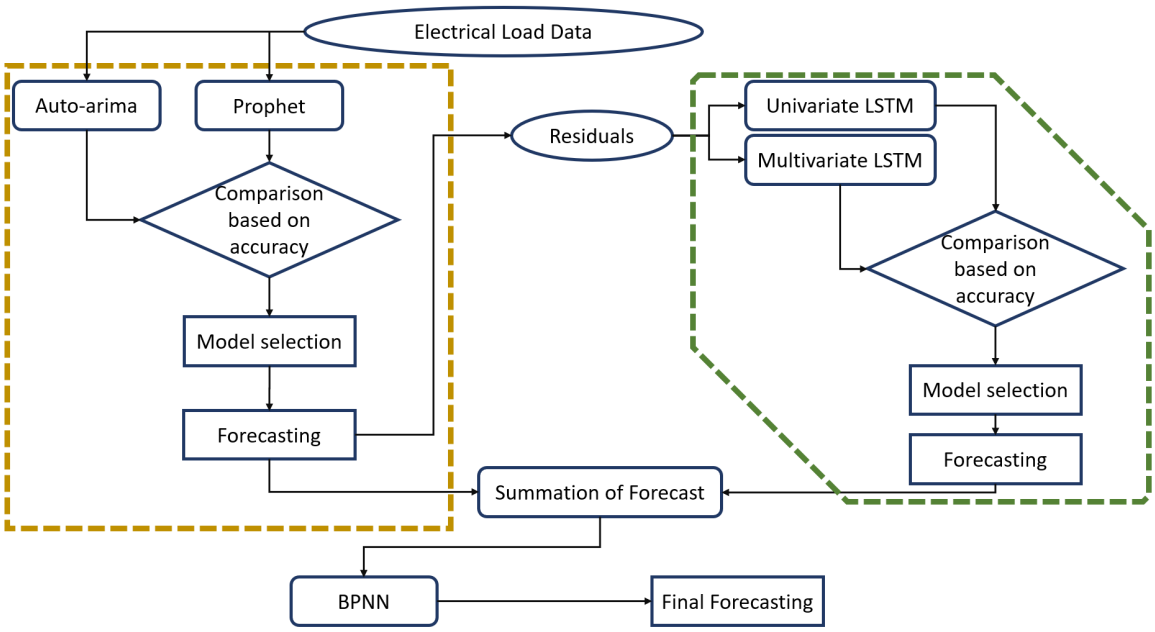


Figura 4: Diagrama de flujo del modelado del modelo híbrido propuesto. El recuadro dorado indica la parte lineal del modelo, el recuadro verde, la parte no lineal.

H. Evaluación

Para tener una comparación del desempeño del modelo híbrido, realizamos el mismo proceso de predicción con cada mejor modelo de cada etapa. Estos son los resultados por etapa:

Modelo	MAPE
Auto-arima (2, 1, 4)	14 %
Prophet	10 %

Cuadro V: Comparación Lineal

Cuadro VI: Errores de medición para cada modelo evaluado en las etapas preliminares al modelo híbrido. MeAE es Median Absolute Error

Error type	Lineal		No lineal	
	Auto-arima	Prophet	LSTM Univariado	**LSTM Multivariado**
MAPE	0.1441	0.4784	1.7366	3.2209
MAE	1200.68	3639.89	980.30	762.22
MSE	2293601.99	16356935.08	1462179.34	1016760.89
MeAE	1008.22	3934.57	845.57	585.61

Cuadro VII: Errores medidos para los datos completos en los diferentes modelos evaluados (Solo se evaluaron los mejores de cada tipo). BPNN se refiere al modelo híbrido

Métrica	Auto-arima	**BPNN**	LSTM Multivariado
MAPE	0.1441	0.0448	0.0961
MAE	1200.68	363.79	758.69
MSE	2293601.99	332969.49	1011745.38
MeAE	1008.22	287.54	579.99

III. ANÁLISIS

De los resultados de las comparaciones lineales, podemos ver que aún siendo Prophet un modelo más complejo que ARIMA y aún añadiendo las consideraciones de holidays en el modelo, ARIMA fue superior en las métricas evaluadas. Nuevamente reafirmando que no siempre los modelos más complejos son los mejores. Entre las comparaciones de la comparación en modelos no lineales, no hay una conclusión clara entre cual escoger, si el multivariado o univariado. Sin embargo, el multivariado tiende a un error más pequeño y además es más robusto, pues utiliza varias características adicionales importantes que consideramos podrían aportar más en la evaluación de los modelos de los datos originales, no los residuales, que se pueden ver en el cuadro VII, donde LSTM multivariado compite bien contra el modelo híbrido, mejor que el modelo lineal.

Finalmente, en este mismo ultimo cuadro, podemos ver que los modelos híbridos si son superiores a sus dos componentes individuales, en donde en todas las métricas el modelo hibrido fue superior a sus dos contrapartes.

IV. CONCLUSIONES

- En la comparación lineal, pudimos encontrar que el modelo de auto-arima, un ARIMAX (2, 1, 4) tuvo mejor resultado en desempeño predictivo que Prophet de Facebook, para la serie de tiempo estudiada.
- En la comparación no lineal, aunque no haya una diferencia muy marcada, la estimación multivariada da un poco de mejor precisión y ayuda mejor a captar modificaciones en la secuencia ayudado por la ingeniería de caracteristicas.
- Encontramos que hay muchas características en donde la ingeniería de estas puede aportar bastante información, así se parta de una seria univariada, el extraer información de los mismos datos (Fourier) resultó aportante en la explicación no lineal del tiempo.
- Los modelos híbridos, por lo menos bajo los parámetros estudiados, son mejores que los modelos tradicionales en poder predictivo para corto o mediano plazo y aportarían un gran beneficio en la aplicación en industria energética.

V. REFERENCIAS

[1] T. Bashir, C. Haoyong, M. F. Tahir, and Z. Liqiang, "Short term electricity load forecasting using hybrid prophet-lstm model optimized by bpnn," *Energy Reports*, vol. 8, pp. 1678–1686, 11 2022.

[2] P. N. Stuart J. Russell, *Artificial Intelligence: A Modern Approach*, 3rd ed. Prentice Hall, 2010.

[3] S. J. Taylor and B. Letham, "Forecasting at scale," *The American Statistician*, vol. 72, pp. 37–45, 1 2018.