



UNIVERSIDAD
TORCUATO DI TELLA

Nowcasting de la actividad económica en
Argentina: una aproximación desde
Machine Learning

Tesis Final

Maestría en Econometría

Alumno: Tobías Pejkovich Balbiani

Legajo: 147275

Tutora: Magdalena Cornejo

Fecha: 06/06/2022

Índice

1. Introducción.....	- 2 -
2. Revisión de la literatura empírica.....	- 4 -
3. Modelos	- 8 -
3.1 ARIMA.....	- 8 -
3.2 VAR.....	- 11 -
3.3 FAVAR.....	- 12 -
3.4 Análisis Espectral Singular.....	- 14 -
3.5 Redes Neuronales recurrentes	- 19 -
3.5.1 Jordan	- 25 -
3.5.2 <i>Long Short Term Memory</i>	- 26 -
4. Base de datos	- 32 -
4.1 Indicadores	- 32 -
4.2 Transformación de los datos.....	- 34 -
5. Estimaciones y resultados.....	- 36 -
5.1 Estimaciones para el conjunto <i>Test</i>	- 37 -
5.1.1 ARIMA.....	- 37 -
5.1.2 VAR.....	- 38 -
5.1.3 FAVAR.....	- 40 -
5.1.4 Análisis Espectral Singular.....	- 41 -
5.1.5 Jordan	- 43 -
5.1.6 <i>Long-Short Term Memory</i>	- 44 -
5.2 Estimaciones para toda la muestra y proyección de enero de 2022	- 48 -
5.3 <i>Test</i> de Diebold-Mariano	- 54 -
5.4 Extensión para modelos multivariados: estimaciones en “tiempo real”.....	- 56 -
6. Conclusiones.....	- 60 -
7. Referencias bibliográficas	- 63 -
8. Anexo	- 67 -

1. Introducción

El Producto Bruto Interno (PBI) constituye el indicador macroeconómico más relevante de cualquier economía. En Argentina se suele publicar con frecuencia trimestral y se da a conocer con 11 semanas de rezago aproximadamente. Adicionalmente, el Instituto Nacional de Estadísticas y Censos (INDEC) elabora, mensualmente, el Estimador Mensual de la Actividad Económica (EMAE), un indicador provisorio de la evolución del PBI. El EMAE se trata de un índice de *Laspeyres* que ofrece una pauta del comportamiento de la actividad económica con una frecuencia mayor a la del PBI trimestral a precios constantes y se publica como un índice mensual con base 100 en el año 2004. Su cálculo se estructura en la agregación del valor agregado a precios básicos de cada uno de los sectores económicos más los impuestos netos de subsidios a los productos; utilizando información parcial, provisorio o alternativa a la del PBI para brindar información anticipada de cómo viene evolucionando la economía. Utilizando las ponderaciones de las cuentas nacionales con base en el año 2004, el indicador se obtiene como la agregación de las series mensuales de las diferentes actividades. Sin embargo, vale la pena destacar que el EMAE también se da a conocer con un rezago considerable, el cual es de 7 semanas aproximadamente.

Debido a la demora con la que se publica la primera estimación oficial de la actividad económica, resulta interesante elaborar una estimación en tiempo “real” o con un menor rezago, lo que se conoce en la literatura como *Nowcasting*. El *Nowcasting*, concepto que combina a los términos *now* (ahora) y *forecasting* (proyección, predicción o pronóstico), ha sido definido como la predicción del presente y/o del pasado muy reciente. Es una técnica desarrollada inicialmente en la meteorología que también ha sido adoptada por la economía, la cual ha demostrado ser una herramienta útil para realizar estimaciones con información parcial adelantada. En definitiva, el objetivo principal del *Nowcasting* es realizar una estimación temprana de una variable de interés, aprovechando la información que se publica de manera anticipada y, en algunos casos, con frecuencia más alta. La principal ganancia de emplear este tipo de técnicas es la posibilidad de contar con un indicador prácticamente en tiempo real, donde las estimaciones pueden ser actualizadas a medida que se conoce la nueva información.

El objetivo de este trabajo es proyectar el EMAE sin estacionalidad (EMAE s.e.) a un mes vista con métodos de aprendizaje supervisado, tanto desde un enfoque univariado como multivariado (utilizando una serie de indicadores económicos que se conocen anticipadamente al EMAE), y contrastarlos con respecto a otros modelos econométricos más tradicionales de series de tiempo. En definitiva, el objetivo de este trabajo es intentar obtener una estimación fiable del EMAE s.e. de forma anticipada a la publicación del dato oficial, por lo que se busca un equilibrio entre celeridad y precisión para medir la evolución de la actividad económica con un menor rezago.

El trabajo está estructurado de la siguiente manera: en el segundo apartado se presenta una revisión de la literatura que resume un conjunto de trabajos que buscan predecir la evolución de la actividad económica con modelos de *Machine Learning*; en la tercera sección se exhiben, desde un enfoque teórico, los diversos modelos que serán utilizados para la estimación del EMAE s.e.; en el cuarto apartado se describen los datos utilizados y sus transformaciones; en la quinta sección se presentan los resultados de las estimaciones y el desempeño relativo de los distintos modelos para un período de “testeo” entre 2017 y 2021, para toda la muestra entre 2005 y 2021 y se evalúan las proyecciones puntuales para enero de 2022; a la vez que se verifica si las predicciones son estadísticamente diferentes entre sí y se realiza una extensión para los modelos multivariados utilizando aquellos indicadores que se conocen en los primeros 20 días posteriores al cierre de cada mes para obtener una estimación del EMAE s.e. en “tiempo real”; y en el sexto apartado se esbozan algunas conclusiones sobre el presente trabajo.

2. Revisión de la literatura empírica

La literatura que hace referencia a las proyecciones de series de tiempo ha desarrollado diversas técnicas para hacer uso del creciente número de indicadores disponibles para predecir una variable de interés. Dentro de ellas, encontramos las Ecuaciones Puente, las Regresiones de Frecuencias Mixtas y los modelos de Factores Dinámicos. Esta última metodología es la que se utiliza actualmente en el Banco Central de la República Argentina (BCRA) para realizar la estimación del PBI mediante ejercicios de *Nowcasting*. De hecho, este trabajo se inspira en una serie de documentos del BCRA¹ en los que se realiza un ejercicio de *Nowcasting* del PBI de Argentina utilizando un amplio conjunto de indicadores para construir modelos que incluyen diferentes versiones de un modelo de Factores Dinámico, Ecuaciones Puente y un *Factor Augmented Vector Regression*. En el primer trabajo publicado, D'Amato, Garegnani & Blanco (2016) encuentran que dichas metodologías brindan resultados superiores a un modelo autoregresivo de orden 1 (AR(1)) y concluyen que poseen capacidades predictivas estadísticamente significativas.

En este trabajo se intenta ampliar el conjunto de modelos utilizados para proyectar la evolución de la actividad económica en Argentina al agregar modelos de aprendizaje supervisado o pertenecientes al universo de *Machine Learning*, tal como el Análisis Espectral Singular o las Redes Neuronales Recurrentes. Algunas instituciones y/o académicos ya han avanzado en este frente al estimar la evolución de la actividad de diferentes economías con modelos de *Machine Learning*.

Desde el *Bank for International Settlements* (BIS), Richardson, Van Florenstein Mulder & Vehbi (2018) evalúan diversos algoritmos de *Machine Learning* para estimar la evolución del PBI de Nueva Zelanda, entre los que encontramos: *K-Nearest Neighbour Regression* (KNN), *Boosted Trees* (BT), *Least Absolute Shrinkage and Selection Operator* (LASSO), *Elastic Nets* (ENET), *Support Vector Machine Regression* (SVM) y *Neural Networks* (NN). Todos estos métodos son comparados con otros modelos econométricos de referencia: un AR(1), un modelo de factores (FM), un *Bayesian Vector Autoregressive* (BVAR) y un conjunto de modelos estadísticos utilizados en el Banco Central de Nueva Zelanda. Para ello, utilizan un amplio conjunto de indicadores

¹D'Amato, Garegnani & Blanco (2016), Blanco, D'Amato, Dogliolo & Garegnani (2017) y BCRA (2020).

económicos y financieros de índole local e internacional y replican un ejercicio de *Nowcasting* entre 2009 y 2018. Los autores encuentran que los modelos SVM, LASSO y NN son los tres mejores métodos de *Machine Learning* ya que reducen entre un 16% y 18% los errores de predicción con respecto al modelo *benchmark* AR(1) y demuestran, mediante el *test* de Diebold-Mariano, que las proyecciones de estos tres modelos son estadísticamente más precisas que las del AR(1). Asimismo, la mayoría de los modelos de *Machine Learning* también tienen un mejor desempeño que el BVAR y el FM. Con todo, los resultados de este trabajo indican que hay ganancias en la precisión de las proyecciones al realizar un ejercicio de *Nowcasting* con modelos de *Machine Learning*, motivo por el cual sus algoritmos constituyen una herramienta útil para predecir la evolución del PBI de Nueva Zelanda.

Desde el Banco Central de Canadá, Tkacz & Hu (1999) utilizan una serie de indicadores financieros y monetarios entre 1968 y 1998 para predecir el PBI a uno y cuatro trimestres vista utilizando Redes Neuronales Artificiales y un conjunto de modelos lineales en los parámetros. Los autores encuentran que las Redes Neuronales producen mejoras significativas en las proyecciones con un horizonte de cuatro trimestres, con un error entre un 15% y 19% menor que los modelos lineales en los parámetros (cifra similar a los resultados del BIS); aunque, con un horizonte de un trimestre, los resultados entre los modelos son estadísticamente similares entre sí. Según Tkacz & Hu (1999), la mejora en la predicción a cuatro trimestres se debe a que las Redes Neuronales capturan relaciones no lineales entre las variables monetarias y financieras y el crecimiento del PBI en el horizonte más largo.

Por su parte, Loermann & Maas (2019) elaboran proyecciones del PBI de Estados Unidos utilizando datos desde 1959 y evalúan el desempeño relativo desde 1999 de una Red Neuronal Artificial al utilizar un perceptrón multicapa o *multi-layer perceptron* (MLP) con respecto a un modelo dinámico de factores (DFM) y una encuesta denominada *Survey of Professional Forecasters* que publica el *Federal Reserve Bank of Philadelphia*. Los resultados indican que el MLP supera al DFM, pero que genera, al menos, tan buenos pronósticos como la *Survey of Professional Forecasters*. Longo, Riccaboni & Rungi (2022) también evalúan, en una parte de su trabajo, el desempeño relativo entre una Red Neuronal Recurrente con un DFM para predecir la evolución del PBI de Estados Unidos, cuyos resultados sugieren que la principal ventaja comparativa de las Redes Neuronales

radica en las proyecciones de corto plazo. En esta línea, Tang, De Almeida & Fishwick (1991) también encuentran que las Redes Neuronales tienen un mejor desempeño relativo que los modelos econométricos más tradicionales de series tiempo para modelos con memoria de corto plazo, mientras que para series con mayor memoria producen resultados similares.

En relación a los países emergentes, Sunitha, Sampath Kumar, Jyothirani & Haragopal (2018) utilizan datos del PBI de India entre 1951 y 2016 y encuentran que un MLP posee un desempeño superior a un *Autoregressive Integrated Moving Average* para proyectar el PBI de India. Utilizando datos entre 1970 y 2015, Sanusi, Moosin & Kusairi (2020) encuentran que un modelo de Redes Neuronales tiene una mejor capacidad predictiva que las estimaciones del gobierno para proyectar la evolución del PBI de Malasia. Además, Olaniyi, Ibina & Eze (2014) utilizan una serie de indicadores financieros entre 1990 y 2009 para predecir el mercado de acciones y, posteriormente, el PBI de Nigeria. En este trabajo, los autores encuentran que una Red Neuronal entrenada con datos suficientes y los *inputs* correctos supera significativamente a los análisis de regresión lineal ya que las cinco métricas que utilizan para medir el error son más bajas para la Red Neuronal.

Al considerar un conjunto de países en un mismo trabajo, Jahn (2018) encuentra que una Red Neuronal Artificial produce predicciones mucho más precisas que un conjunto de modelos lineales para la estimación de la tasa de crecimiento anual del PBI de 15 economías industrializadas (Alemania, Austria, Bélgica, Dinamarca, España, Estados Unidos, Finlandia, Francia, Grecia, Holanda, Italia, Japón, Portugal, Reino Unido y Suecia) entre 1996 y 2016. En particular, el autor destaca que las Redes Neuronales pueden capturar tendencias temporales de forma muy flexible. En cambio, Heravi, Osborn & Birchenhall (2004) consideran 24 series que miden la variación anual de la producción industrial de diversos sectores de Alemania, Francia e Inglaterra; las cuales muestran poca evidencia de no linealidad. En este marco, los autores sostienen que los modelos lineales en los parámetros producen proyecciones hasta a 1 año más precisas que las Redes Neuronales, pero que estos últimos dominan al momento de predecir el signo de la variación anual. Chuku, Oduor & Simpasa (2017) analizan los datos del PBI de Sudáfrica, Nigeria y Kenia entre 1970 y 2016 y encuentran que, en muchas de las especificaciones consideradas, el modelo de Redes Neuronales es algo superior a los

modelos econométricos estructurales y a los modelos *Autoregressive Integrated Moving Average*; especialmente cuando los precios de *commodities*, el comercio, la inflación y la tasa de interés se incluyen como variables explicativas.

En relación al Análisis Espectral Singular, Hassani, Rua, Silva & Thomakos (2019) trabajan con series de distinta frecuencia al utilizar datos de PBI y producción industrial entre 1960 y 2013 para 11 países desarrollados (Austria, Bélgica, Dinamarca, Estados Unidos, Francia, Holanda, Italia, Japón, Portugal, Reino Unido y Suecia). Al obtener estimaciones mensuales del PBI, los autores encuentran que el Análisis Espectral Singular Multivariante permite lograr ganancias de pronóstico notables para todos los países.

En este marco, este trabajo busca, al igual que los documentos mencionados, evaluar si los modelos de *Machine Learning* son herramientas útiles para predecir la evolución de la actividad económica de Argentina en el corto plazo, cuyos resultados son contrastados con modelos econométricos más tradicionales de series de tiempo.

3. Modelos

La proyección del EMAE s.e. se realizará con diferentes modelos de aprendizaje supervisado, los cuales comprenden al Análisis Espectral Singular o *Singular Spectrum Analysis* (SSA) y a las Redes Neuronales Recurrentes o *Recurrent Neural Networks* (RNN). Con respecto al primer modelo, se utilizará un enfoque univariado; mientras que, en relación a los segundos, se trabajará con dos modelos particulares: la Red Neuronal de Jordan y el modelo *Long Short Term Memory* (LSTM). A su vez, estos modelos de *Machine Learning* serán contrastados con tres modelos econométricos más tradicionales: un *Autoregressive Integrated Moving Average* (ARIMA), un *Vector Autoregression* (VAR) y un *Factor Augmented Vector Autoregression* (FAVAR). A continuación, se presentan los principales lineamientos teóricos de todos los modelos, haciendo mayor énfasis en los de *Machine Learning*.

Tabla 1. Resumen de los modelos utilizados para proyectar el EMAE s.e.

	Univariado	Multivariado
Modelos Econométricos	ARIMA	VAR, FAVAR
Machine Learning	SSA y LSTM	Jordan y LSTM

Fuente: elaboración propia

3.1 ARIMA

Un *Autoregressive Moving Average* (ARMA) es un modelo que combina procesos autoregresivos (AR) de orden p y de medias móviles (MA) de orden q . La variable Y_t representa la serie temporal de interés, ϕ_p representa a los parámetros del componente AR, θ_q representa a los parámetros del componente MA, mientras que el término de error ε_t es ruido blanco, es decir, tiene media cero, varianza constante y está incorrelacionado con respecto a su propio estado.

$$AR: Y_t = \phi_0 + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \dots + \phi_p Y_{t-p} = \phi_0 + \sum_{i=1}^p \phi_i Y_{t-i} \quad (1)$$

$$MA: Y_t = \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \dots - \theta_q \varepsilon_{t-q} = \varepsilon_t - \sum_{i=1}^q \theta_i \varepsilon_{t-i} \quad (2)$$

$$ARMA: Y_t = \phi_0 + \sum_{i=1}^p \phi_i Y_{t-i} + \varepsilon_t - \sum_{i=1}^q \theta_i \varepsilon_{t-i} \quad (3)$$

Un ARIMA no es más que un modelo ARMA aplicado a una serie temporal integrada de orden d . Se trata de una serie temporal que se tiene que diferenciar d veces para que se convierta en una serie estacionaria, es decir, que al ser diferenciada d veces se transforma en un ARMA(p, q). Por lo tanto, los parámetros p , d y q determinan los órdenes del componente autoregresivo, de integración y de media móvil respectivamente en un modelo ARIMA(p, d, q).

$$ARIMA: \Delta^d Y_t = \sum_{i=1}^p \phi_i \Delta^d Y_{t-i} + \varepsilon_t - \sum_{i=1}^q \theta_i \varepsilon_{t-i} \quad (4)$$

$$ARIMA: Y_t = -(\Delta^d Y_t - Y_t) + \sum_{i=1}^p \phi_i \Delta^d Y_{t-i} + \varepsilon_t - \sum_{i=1}^q \theta_i \varepsilon_{t-i} \quad (5)$$

El principio en el que se basa la estimación de los parámetros de un modelo ARMA(p, q) o ARIMA(p, d, q) es el de máxima verosimilitud. En la práctica, la verosimilitud se maximiza mediante procedimientos numéricos. Sin embargo, cuando estimamos dichos modelos, por lo general evaluamos la máxima verosimilitud condicional. El término condicional hace referencia a que se asume que se conocen las primeras $\max(p, q)$ observaciones. Asintóticamente, la estimación vía máxima verosimilitud condicional es equivalente a la estimación vía máxima verosimilitud y más fácil de calcular.

A la hora de seleccionar el modelo ARMA o ARIMA adecuado, se sigue la metodología de *Box-Jenkins*, la cual consiste en una serie de pasos. En primer lugar, se tiene que asegurar que la serie temporal de interés sea estacionaria, lo cual se puede llevar a cabo mediante un test de *Dickey-Fuller* para un modelo ARMA o determinando el orden de integración d para un modelo ARIMA. Luego, para elegir el orden p y q correctos, se suele analizar dos resultados: la significatividad estadística de los coeficientes de los parámetros estimados y las funciones de autocorrelación (ACF) y de autocorrelación parcial (PACF) con el objetivo de verificar si los residuos poseen (o no) estructura. En el

primer caso, un p-valor menor a 0,05 indica que el coeficiente del parámetro estimado es estadísticamente significativo ya que se rechaza la hipótesis nula de que $\phi_i = 0$ para el componente AR o de que $\theta_i = 0$ para el componente MA. En el segundo caso, un p-valor mayor a 0,05 indica que los residuos no son estadísticamente diferentes de cero ya que no se rechaza la hipótesis nula de que $\varepsilon_t = 0$. Una vez hecho esto, si más de un modelo ARIMA o ARMA estimado satisface estas condiciones, se seleccionará aquel que brinde el menor criterio de información. Los criterios de información más conocidos son el de Akaike (AIC) y el de Schwarz (BIC). Para un modelo ARMA(p,q) dichos criterios de información poseen la siguiente forma:

$$AIC = \log(\hat{\sigma}^2) + \frac{2(p+q)}{T} \quad (6)$$

$$BIC = \log(\hat{\sigma}^2) + \frac{2(p+q)}{T} \log(T) \quad (7)$$

Donde $\hat{\sigma}^2$ es la varianza de los residuos, p es el orden del componente AR, q es el orden del componente MA y T es la cantidad de observaciones temporales.

El BIC elige modelos más parsimoniosos y tiene mejores propiedades para muestras grandes. En cambio, el AIC selecciona modelos sobre-parametrizados cuando la muestra tiende a infinito, pero suele superar al BIC bajo muestras pequeñas. Si los dos criterios eligen especificaciones diferentes, con la especificación que elige el BIC se debe verificar la estructura de los residuos y con la especificación que elige el AIC se debe verificar si los parámetros son estadísticamente significativos.

Los modelos ARIMA constituyen una herramienta apropiada para proyectar una serie temporal de interés ya que suelen tener un buen desempeño para realizar proyecciones de corto plazo utilizando solamente valores rezagados de dicha variable. Se trata de un modelo paramétrico y parsimonioso que incluye un amplio conjunto de casos particulares: AR(p), MA(q) y ARMA(p, q). En este punto, vale la pena recordar que el Teorema de Wold indica que un proceso estacionario en sentido débil, es decir que su media y varianza existen y no dependen del tiempo, puede ser escrito como un proceso ARMA(p, q). Además de ello, un modelo ARIMA es capaz de modelar series no

estacionarias producto del componente de integración. La principal debilidad de estos modelos radica en que no incorporan la información que podrían aportar otras variables que se vinculan con la serie temporal de interés y que solo utilizan información del pasado para predecir el futuro.

3.2 VAR

Un VAR es un proceso autoregresivo para un vector de variables, el cual comprende una ecuación por cada variable del sistema, donde el parámetro p determina el orden del componente autoregresivo. En este caso, la variable Y_t hace referencia a la serie temporal de interés, mientras que X_t representa el resto de las $k - 1$ series temporales que se incluyen dentro del vector de variables. Y_t y X_t están agrupadas como variables endógenas en las matrices W ; las matrices A contienen los parámetros del modelo VAR y el término de error ε_t también es ruido blanco, es decir, tiene media cero, varianza constante y está incorrelacionado con respecto a su propio estado.

$$W_t = A_1 W_{t-1} + A_2 W_{t-2} + \dots + A_p W_{t-p} + \varepsilon_t = \sum_{i=1}^p A_i W_{t-i} + \varepsilon_t \quad (8)$$

$$W_t = \begin{bmatrix} Y_t \\ X_{1t} \\ \vdots \\ X_{(k-1)t} \end{bmatrix}; A_{k \times k} = \begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} & \dots & \phi_{1k} \\ \phi_{21} & \phi_{22} & \phi_{23} & \dots & \phi_{2k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi_{k1} & \phi_{k2} & \phi_{k3} & \dots & \phi_{kk} \end{bmatrix}; \varepsilon_t = \begin{bmatrix} \varepsilon_{1t} \\ \varepsilon_{2t} \\ \vdots \\ \varepsilon_{kt} \end{bmatrix} \quad (9)$$

Al igual que los modelos ARIMA, los modelos VAR también se estiman mediante máxima verosimilitud. En la práctica, para estimar un modelo VAR también se tiene que comprobar en primer lugar que las series temporales en cuestión son estacionarias. Para seleccionar el orden p adecuado, se puede elegir el número óptimo de rezagos en base al *test* de ratio de la verosimilitud (LR *test*) o en base al menor resultado de los diferentes criterios de información. En términos generales, para analizar la estructura asociada al proceso de generación de datos, se suele mirar el LR *test*; mientras que, para proyectar una serie temporal de interés, se suele mirar los criterios de información.

En definitiva, un modelo VAR es útil cuando se busca predecir múltiples variables de series de tiempo utilizando un solo modelo o cuando busca incorporar las asociaciones entre la serie temporal de interés y otras variables explicativas. Además, los modelos

VAR tienen una serie de herramientas adicionales que pueden resultar útiles. En particular, con este método se puede (i) comprender si determinadas variables ayudan a predecir a la serie temporal de interés mediante un *test* de causalidad a la Granger, (ii) ver cómo responde una variable determinada ante un *shock* en otra variable mediante el análisis de las funciones impulso-respuesta y (iii) estimar la incidencia de otras variables explicativas sobre la varianza de la serie temporal de interés mediante un análisis de la descomposición de la varianza, la cual estará determinada por el ordenamiento de Cholesky establecido. Sin embargo, los modelos VAR requieren que las series sean estacionarias y no permiten incorporar demasiadas variables ya que, en esos casos, tienen que estimar una gran cantidad de parámetros; a la vez que los residuos en la forma reducida no tienen ninguna interpretación económica y el ordenamiento de Cholesky suele ser arbitrario.

3.3 FAVAR

El modelo VAR suele ser un buen método de estimación para proyectar, pero desafortunadamente puede tener un pobre desempeño cuando hay numerosas variables debido a que tiene que estimar una gran cantidad de parámetros y, como sucede en este trabajo, podríamos no disponer de los grados de libertad suficientes. Los dos métodos usuales para lidiar con el problema de dimensionalidad de los datos suelen ser un VAR bayesiano (BVAR) o un modelo de factores dinámico (DFM). Para lidiar con esta problemática en este trabajo, se estima un FAVAR.

Un modelo FAVAR es un método que reduce la dimensionalidad de los datos estimando factores subyacentes al conjunto de variables, los cuales se utilizan posteriormente en un VAR. Por lo tanto, un FAVAR básicamente combina un modelo VAR con un modelo DFM. El concepto subyacente detrás de los DFM radica en que la evolución de las variables puede ser resumida en un pequeño número de factores latentes (los cuales pueden ser considerados como un resumen exhaustivo de la información que contienen los datos) más una serie de *shocks* idiosincráticos. En términos simplificados, un FAVAR permite resumir la información latente o no observable contenida en un amplio conjunto de indicadores económicos en unos pocos factores comunes. Por lo tanto, un vector de N variables observables puede ser explicado por los rezagos distribuidos de K factores comunes más N perturbaciones idiosincráticas que eventualmente podrían

estar correlacionadas. Utilizando este enfoque, más información puede ser incorporada en relación a un modelo VAR estándar y, consecuentemente, se aborda la problemática del sesgo de variables omitidas.

Un DFM en su forma estática puede presentarse de la siguiente manera:

$$Y_t = \Lambda F_t + D(L)Y_{t-1} + v_t \quad (10)$$

$$F_t = \Phi(L)F_{t-1} + G\eta_t \quad (11)$$

Y_t es un vector de variables observables de tamaño $M \times 1$, F_t es un vector de factores no observables de tamaño $K \times 1$, Λ es una matriz de $M \times K$ y G es una matriz de $K \times N$. La primera ecuación se denomina ecuación medición y la segunda se denomina ecuación estado. Esta representación es denominada estática ya que F_t aparece en la ecuación de medición sin rezagos. Reemplazando la segunda ecuación en la primera, es posible escribir el DFM en un modelo VAR:

$$\begin{bmatrix} F_t \\ Y_t \end{bmatrix} = \begin{bmatrix} \Phi(L) & 0 \\ \Lambda \Phi(L) & D(L) \end{bmatrix} \begin{bmatrix} F_{t-1} \\ Y_{t-1} \end{bmatrix} + \begin{bmatrix} \varepsilon_{F_t} \\ \varepsilon_{X_t} \end{bmatrix} \quad (12)$$

$$\begin{bmatrix} \varepsilon_{F_t} \\ \varepsilon_{X_t} \end{bmatrix} = \begin{bmatrix} I \\ \Lambda \end{bmatrix} G\eta_t + \begin{bmatrix} 0 \\ v_t \end{bmatrix} \quad (13)$$

$$X_t = \Lambda^f F + \Lambda^y Y_t + e_t \quad (14)$$

X_t es un vector de $N \times 1$ de series temporales, Y_t es un vector de $M \times 1$ de series temporales observables que representan un subconjunto de X_t . Λ^f es la matriz de cargas de tamaño $N \times K$, Λ^y es la matriz de coeficientes que une a las variables observables Y_t con el resto de las series de tamaño $N \times M$ y e_t es el vector de errores de tamaño $N \times 1$, los cuales se encuentran normalmente distribuidos con media cero y varianza constante.

Para la estimación del modelo FAVAR se emplea un análisis de componentes principales en dos etapas. En el primer paso, se identifican los factores comunes \hat{F}_t extraídos de Y_t y X_t . Los factores pueden ser estimados utilizando los primeros R

componentes principales de X , siempre y cuando la cantidad estimada de factores K sea superior a R . En el segundo paso, se estima la ecuación de estado reemplazando a F_t por \hat{F}_t .

3.4 Análisis Espectral Singular

El SSA es una metodología de análisis y pronóstico de series temporales con un amplio abanico de aplicaciones, el cual va desde la descomposición y filtración de las series hasta la estimación de parámetros y la elaboración de proyecciones. Si bien en el modelo SSA se emplean algunos conceptos probabilísticos y estadísticos, la principal diferencia con respecto a la mayoría de los métodos econométricos de series de tiempo radica en que no es necesario hacer ningún supuesto estadístico con respecto a los datos, tal como la estacionariedad de las series o la normalidad de los residuos. Otro aspecto relevante para aplicaciones económicas es que este modelo funciona bien en muestras pequeñas y es robusto a *outliers*.

El modelo SSA consta de dos etapas complementarias, las cuales incluyen dos pasos separados cada una. En términos generales, el modelo SSA descompone a la serie temporal en un conjunto de componentes independientes e interpretables (tendencia, ciclo, estacionalidad, componente aleatorio) y reconstruye la serie original (ignorando el componente aleatorio) para proyectar nuevos puntos de datos. La primera etapa es la de descomposición, donde la serie temporal se transforma en una matriz de trayectorias y luego se aplica la descomposición en valores singulares a dicha matriz de trayectorias para obtener una descomposición en componentes elementales de matriz de rango uno. La segunda etapa es la de reconstrucción, la cual crea componentes agrupados identificando las señales y el componente aleatorio y luego reconstruye la nueva serie sin el componente aleatorio mediante el promedio diagonal o la “Hankelización”. A continuación, se presenta una breve guía con las 2 etapas y los 4 pasos del modelo SSA.

Considerando una serie temporal $Y_T = (y_1, \dots, y_T)$, se fija la longitud de la ventana temporal L , siendo este un número entero tal que $2 \leq L \leq T$, con $K = T - L + 1$

Paso 1: Incrustación. La serie temporal es transformada en una matriz de trayectorias mediante un operador de incrustación. La incrustación puede ser considerada

como un mapeo que transfiere una serie temporal univariada $Y_T = (y_1, \dots, y_T)$ a un conjunto de series multivariadas X_1, \dots, X_K , donde $X_i = (y_i, \dots, y_{i+L-1})^T$. El resultado de este paso es la matriz de trayectorias $X = [X_1, \dots, X_K]$:

$$X = \begin{bmatrix} y_1 & y_2 & y_3 & \cdots & y_K \\ y_2 & y_3 & y_4 & \cdots & y_{K+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y_L & y_{L+1} & y_{L+2} & \cdots & y_T \end{bmatrix} \quad (15)$$

En ese punto vale la pena notar que la matriz de trayectorias X es una matriz de Hankel².

Paso 2: Descomposición en valores singulares o *Singular Value Decomposition* (SVD). Desde X , se define a la matriz XX^T . En este paso, la matriz de trayectorias se descompone en una suma de matrices elementales de rango uno. La SVD de XX^T produce una cantidad de L autovalores $\lambda = (\lambda_1, \dots, \lambda_L)$ y L autovectores $P = (P_1, \dots, P_L)$ de la matriz XX^T y la representa en la forma de $XX^T = P \Lambda P^T$, donde $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_L)$ es la matriz diagonal de los autovalores de XX^T ordenados de forma tal que $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_L \geq 0$ y $P = (P_1, P_2, \dots, P_L)$ es la matriz ortogonal de autovectores de XX^T . Luego se establece $d = \max(i, \text{tal que } \lambda_i > 0) = \text{rango } X$, que representa el número de autovalores mayores a cero o, de forma equivalente, al rango de la matriz de trayectorias. En la práctica, usualmente tenemos $d = L^*$ con $L^* = \min\{L, K\}$.

Si denotamos $V_i = X^T P_i / \sqrt{\lambda_i}$ ($i = 1, \dots, d$), entonces la SVD de la matriz de trayectorias puede ser escrita como:

$$X = X_1 + \cdots + X_d \quad (16)$$

Las matrices X_i tienen rango 1. El conjunto $(\sqrt{\lambda_i}, P_i, V_i)$ se denomina la i -ésima “auto-triples” de la matriz X . La terminología estándar denomina a $\sqrt{\lambda_i}$ como los valores

² Denotando a la letra i como las filas de una matriz y a la letra j como las columnas, una matriz de Hankel es una matriz donde todos los elementos a través de las posiciones $i + j = \text{constantes}$ son iguales. Por ejemplo, en una matriz de Hankel los elementos (3,1), (2,2) y (1,3) son iguales.

singulares de la matriz X , al conjunto $\sqrt{\lambda_i}$ como el espectro de la matriz X , mientras que P_i (denominados funciones ortogonales empíricas del factor) y V_i (denominados componentes principales) son los autovectores a izquierda y derecha de la matriz X respectivamente.

Paso 3: Agrupamiento. El propósito de este paso es la separación aditiva de los componentes de la serie temporal, para lo que se seleccionan M ($1 \leq M \leq L$) autovectores $P_{i_1}, P_{i_2}, \dots, P_{i_M}$. Esto se logra dividiendo las matrices elementales X_i en p grupos y sumando las matrices por cada grupo p . Expresado en fórmulas, sea $I = \{i_1, \dots, i_p\}$, entonces la matriz X_I correspondiente al grupo I se define como $X_I = X_{i_1} + \dots + X_{i_p}$. X_I se asocia a la(s) señal(es) de la serie temporal, mientras que el resto de los $(d - M)$ “auto-triples” representan el componente aleatorio. El conjunto de índices $J = 1, \dots, d$ en los subconjuntos disjuntos I_1, \dots, I_M corresponde a la representación:

$$X = X_{I_1} + \dots + X_{I_M} \quad (17)$$

El procedimiento de elección de los conjuntos I_1, \dots, I_m se denomina agrupación de los “auto-triples”. Para un grupo I dado, la contribución del componente X_i se mide por la proporción de los autovalores correspondientes $\sum_{i \in I} \lambda_i / \sum_{i=1}^d \lambda_i$.

Paso 4: Promedio diagonal o “Hankelización”. En este paso, las matrices agrupadas se transforman a la forma del objeto de entrada (es decir de matrices a series de tiempo) mediante la “Hankelización” o promedio diagonal, donde la idea básica es transformar X_{I_1}, \dots, X_{I_M} en series temporales reconstruidas y_{I_1}, \dots, y_{I_M} . El promedio diagonal transfiere cada matriz X_{I_k} en una serie temporal, la cual es un componente o señal de la serie original. Para ello, se computa la matriz $\tilde{X} = \|\tilde{x}_{ij}\| = \sum_{k=1}^M P_{ik} P_{ik}^T X$ como una aproximación de X .

$$X = \tilde{X}_{I_1} + \dots + \tilde{X}_{I_M} \quad (18)$$

La transición a la serie univariada puede ser alcanzada promediando la diagonal de la matriz \tilde{X} .

Todo esto es equivalente a la descomposición de la serie temporal original $Y_T = (y_1, \dots, y_T)$ en una suma de M series:

$$y_t = \sum_{k=1}^M \tilde{y}_t^{(k)} \quad (19)$$

donde $\tilde{Y}_t^{(k)} = (\tilde{y}_1^{(k)}, \dots, \tilde{y}_T^{(k)})$ corresponde a la matriz X_{Ik}

La clave del modelo SSA radica en la selección de los hiper-parámetros para la descomposición y reconstrucción adecuada de la serie temporal de interés. En este aspecto, el concepto de separabilidad, el cual alude a qué tan bien se pueden separar los diferentes componentes de una serie, juega un rol clave. De hecho, la descomposición solo puede tener éxito si los componentes aditivos de la serie son aproximadamente separables entre sí. Para cuantificar el grado de separabilidad entre dos componentes reconstruidos puede utilizarse la w -correlación ponderada. En términos más formales, sean X_A y X_B dos componentes. Si la correlación $w(X_A, X_B)$ se acerca a cero, las dos series son separables. En cambio, si la correlación $w(X_A, X_B)$ se acerca a uno, los componentes podrían agruparse.

Como se mencionó anteriormente, solo se requieren dos hiper-parámetros en el modelo SSA: la longitud de la ventana L y el número de componentes M . No hay recomendaciones estrictas o reglas universales para su selección, ya que los valores óptimos de estos parámetros dependen de la serie temporal de interés. A pesar de ello, se pueden realizar algunos comentarios generales al respecto.

En relación al parámetro L de la etapa de descomposición, su valor tiene que ser lo suficientemente grande para que cada vector L veces rezagado incorpore una parte del comportamiento de la serie original. De hecho, en la mayoría de los casos se deben elegir valores “elevados” para lograr una mejor separabilidad ya que, a mayor L , mayor es el detalle de la descomposición de la serie temporal. De hecho, el uso de un L pequeño podría conducir a una combinación de componentes que, estando separados, serían interpretables.

En particular, algunos resultados teóricos muestran que es mejor elegir un valor de L en torno a $0,33T - 0,40T$ (Golyandina, 2010). Además, si se conoce que la serie temporal posee un componente estacional periódico, también es recomendable que el parámetro L sea proporcional a ese período para obtener una mejor separabilidad. Por ejemplo, en caso de series con frecuencia mensual, L debería ser igual a un múltiplo de 12. También vale la pena remarcar que variaciones en L pueden influir en las características de separabilidad, aunque los resultados suelen ser estables con respecto a pequeños cambios de L si su valor es relativamente grande.

Para una serie temporal con estructura compleja, la separabilidad de los componentes se torna más difusa y valores demasiado grandes de L pueden conducir a una descomposición no deseada, por lo que se recomienda un valor más pequeño. Por lo tanto, un escenario de estructuras complejas dificulta la selección de un valor adecuado de L ya que una reducción significativa también puede conducir a una mala calidad de la separación, tal como se comentó anteriormente.

En relación al parámetro M , no hay criterios definidos en base a simulaciones como en el caso anterior, sino que más bien dependen de criterios *ad-hoc*. Un enfoque general se basa en la contribución de cada componente a la varianza de la variable de interés evaluada como $\lambda_i / \sum_{i=1}^d \lambda_i$. En este caso, se seleccionan M de d componentes para que la suma de sus contribuciones alcance un umbral predefinido, por ejemplo, el 80%. Es conveniente que dicho umbral tenga un valor relativamente elevado ya que los componentes aleatorios suelen tener una baja contribución. Además, el valor óptimo del parámetro M depende del horizonte de proyección. Un valor bajo para el parámetro M suele funcionar mejor para proyecciones de largo plazo ya que solo interesa capturar unos pocos componentes de baja frecuencia que muestren la tendencia de la serie, mientras que un valor alto suele funcionar mejor para proyecciones de corto plazo.

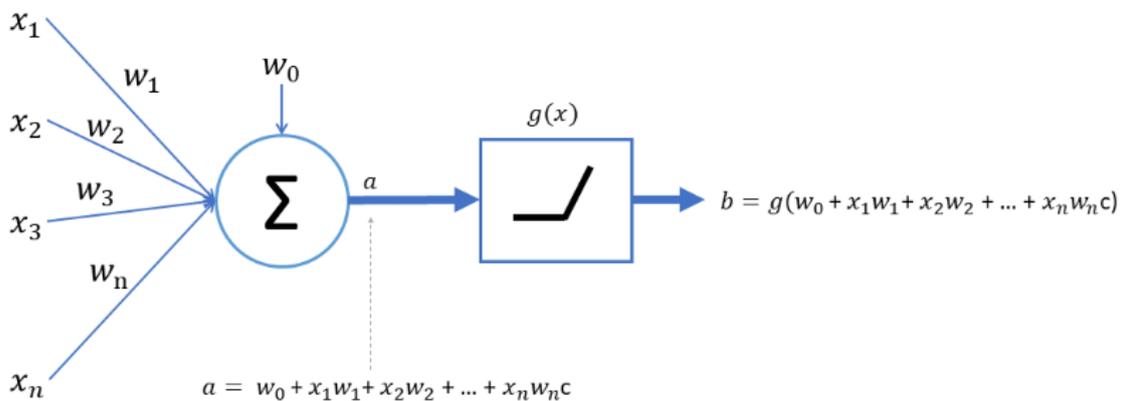
Con todo, se recomienda estimar el modelo SSA en reiteradas oportunidades utilizando diferentes valores de L y M . En particular, en este trabajo se armará una grilla con distintas combinaciones de L y M y se seleccionará aquel modelo que brinde el menor error cuadrático medio en las proyecciones a un mes vista. Sin embargo, vale la pena aclarar que la elección adecuada de L y M según el menor error cuadrático medio no es

necesariamente la mejor, ya que, la estabilidad con respecto a pequeños cambios de L puede ser más importante. En estos casos, para verificar la estabilidad de los pronósticos, los intervalos de confianza pueden ser útiles.

3.5 Redes Neuronales recurrentes

El modelo de Jordan y el modelo LSTM son especificaciones particulares de Redes Neuronales Recurrentes, por lo que comenzaré considerando a las Redes Neuronales en términos generales. Una Red Neuronal Artificial es un modelo que se inspira en el modo en que el cerebro humano procesa la información a través de las neuronas biológicas. Está conformada por un conjunto de unidades de procesamiento denominadas neuronas artificiales, las cuales se encuentran conectadas entre sí y transforman los datos de entrada o *inputs* en datos de salida u *outputs*. La información que ingresa recorre a la Red Neuronal a través de ellas, la cual es transformada en cada neurona hasta alcanzar el final de la red, instancia en la que se brinda un resultado u *output* final.

Gráfico 1. Estructura de una neurona artificial



Fuente: InteractiveChaos.

En mayor detalle, cada neurona recibe un vector de *inputs* $x = (x_1, x_2, x_3, \dots, x_n)$ y los multiplica con sus respectivos pesos o ponderaciones $w = (w_1, w_2, w_3, \dots, w_n)$. La neurona suma los *inputs* ponderados y le agrega un término w_0 que recibe el nombre de sesgo. Al principio las ponderaciones son aleatorias y luego van cambiando gradualmente mediante el aprendizaje. El resultado de la aplicación de esta función lineal se representa en el Gráfico 1 mediante la letra a . A este resultado obtenido se le aplica

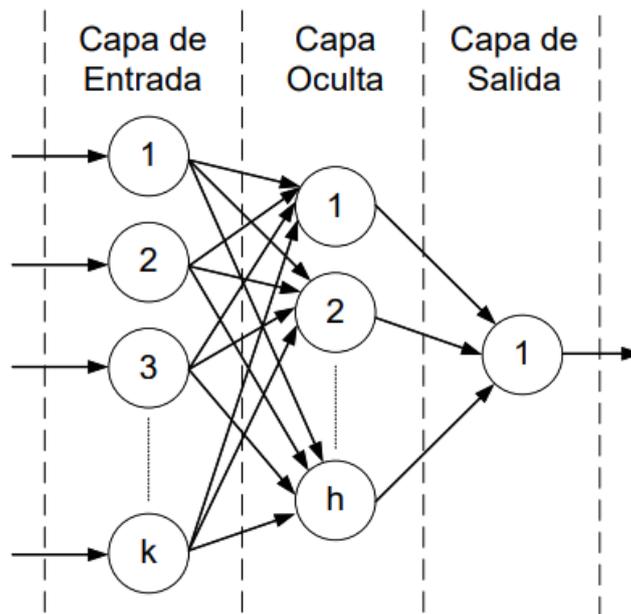
entonces una función $g(x)$ que se denomina función de activación, la cual suele ser no lineal. El *output* final de la neurona es $b = g(w_0 + \sum_{i=1}^n w_i x_i)$.

Tal como se puede ver en el Gráfico 1, las neuronas de una Red Neuronal se conectan mediante pesos o ponderaciones y se organizan en capas. Por un lado, un peso con un valor positivo indica una contribución positiva de la información entre dos neuronas, mientras que un peso negativo marca una incidencia negativa. En este marco, el sesgo es un parámetro adicional que cada neurona puede tener además de sus pesos, el cual desplaza el *output* de la función lineal al agregar una constante. Esto permite un ajuste más preciso, por lo que debe interpretarse como una constante agregada antes de que el *input* ponderado pase a su función de activación.

Por otro lado, las capas suelen ser diferenciadas en tres niveles según su funcionalidad: de entrada, ocultas y de salida. En la capa de entrada las neuronas reciben el vector de *inputs* externo x y propagan el *output* b a la siguiente capa. En la(s) capa(s) oculta(s) las neuronas toman las señales enviadas por la capa anterior (la primera capa oculta recibe las señales de la capa de entrada), también realizan un procesamiento no lineal y envían *outputs* intermedios a la siguiente capa. Los modelos que poseen más de una capa oculta se denominan Redes Neuronales Profundas. Por último, la capa de salida recibe *inputs* de la última capa oculta y proporciona el *output* final del modelo.

En cuanto a las arquitecturas de las Redes Neuronales, el perceptrón posee la estructura más sencilla ya que consta de una o varias capas de entrada y una de salida, pero no posee capas ocultas. En cambio, los MLP son una extensión del perceptrón y se caracterizan por lo siguiente: poseen al menos una capa oculta, cada neurona está conectada con todas las neuronas de la siguiente capa, la información circula de izquierda a derecha y el entrenamiento de la red se suele realizar mediante propagación hacia atrás (*back-propagation*). En el Gráfico 2 se presenta una estructura genérica de una Red Neuronal con una capa de entrada de k neuronas, una capa oculta de h neuronas y una capa de salida de una neurona.

Gráfico 2. Arquitectura de una Red Neuronal



Fuente: Sepúlveda Pinto (2011).

El entrenamiento de una Red Neuronal se refiere a la actualización de los pesos y sesgos y consta principalmente de tres etapas. En primer lugar, el modelo hace una predicción. En segundo lugar, compara dicha predicción con el dato original utilizando una función de pérdida, la cual genera un valor de error. Por último, usa ese valor de error para calcular los gradientes de cada neurona de la red. El gradiente representa el cambio en la función de pérdida como respuesta a un cambio en los parámetros de la Red Neuronal, es decir, es un vector de derivadas parciales que contiene los valores que se utilizan para actualizar las ponderaciones. De esta forma, el gradiente permite el aprendizaje de la red. Cuanto mayor sea el gradiente, mayores serán las magnitudes de los ajustes para disminuir el error de predicción y viceversa. Este proceso se repite muchas veces y la red sigue mejorando sus predicciones hasta alcanzar un punto en el que un nuevo cambio en las ponderaciones no reduce significativamente el error de estimación o hasta alcanzar una cantidad determinada de iteraciones, lo que suceda primero. De esta forma, el entrenamiento de una Red Neuronal denota la búsqueda del error mínimo total, es decir, la selección del conjunto de pesos que minimiza el error.

En particular, las Redes Neuronales actualizan los pesos “viejos” restándoles una fracción de sus respectivos gradientes. Dicha fracción se denomina tasa de aprendizaje,

un valor que se ubica entre 0 y 1 que especifica cuánto cambian los parámetros (ponderaciones y sesgos) cada vez que se actualizan.

$$\text{Nuevo peso} = \text{Viejo peso} - \text{Gradiente} * \text{Tasa de aprendizaje} \quad (20)$$

En mayor detalle, el algoritmo más utilizado para calcular el gradiente es la propagación atrás o *back-propagation*, método más popular para realizar tareas de aprendizaje supervisado. El aprendizaje vía *back-propagation* calcula los gradientes de la función de pérdida analíticamente usando la regla de la cadena desde la capa de salida hacia la capa de entrada de una Red Neuronal. Además, el algoritmo reduce los pesos innecesarios a cero y los elimina durante el entrenamiento. Para implementarlo, se necesita una función de activación continua, no lineal, monotónicamente creciente y diferenciable. La función de activación es aplicada al final con el objetivo de escalar el *output* de la neurona a un valor deseado, usualmente en el rango (0,1) o (-1,1), antes de que sea propagado a la siguiente capa. Las tres funciones de activación más utilizadas son la función de umbral, la función logística y la función de tangente hiperbólica. Todas ellas son funciones de rango acotado, lo que garantiza un buen desempeño de aprendizaje de la red.

$$\text{función de umbral: } \phi(x) = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases} \quad (21)$$

$$\text{función logística: } \phi(x) = \frac{1}{1 + e^{-\beta x}} \quad (22)$$

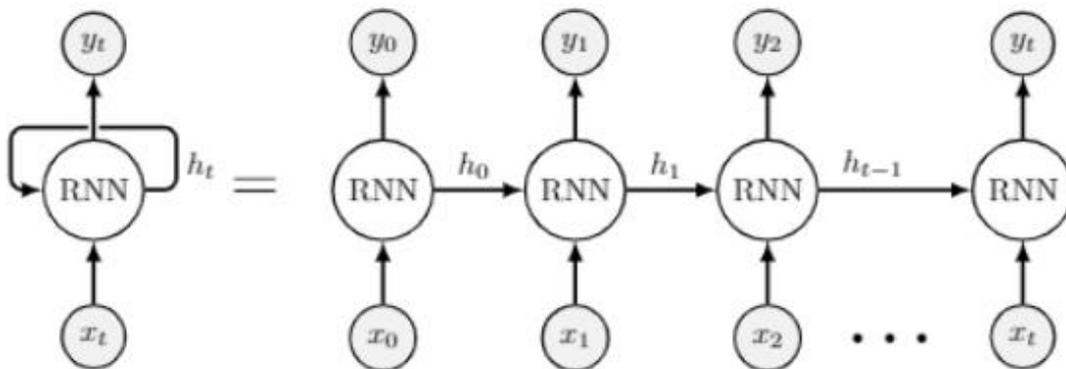
$$\text{función de tangente hiperbólica: } \phi(x) = \tanh(\beta x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (23)$$

En definitiva, la diferencia entre el valor observado y el valor estimado se utiliza para actualizar las ponderaciones y encontrar los parámetros óptimos de la red. En sus aplicaciones prácticas, las Redes Neuronales pueden aprender funciones arbitrariamente complejas (son modelos no lineales en los parámetros) y suelen proporcionar buenas predicciones para problemas de aprendizaje automático. Sin embargo, el MLP es un modelo estático y es incapaz de procesar información temporal. Para que adquieran dinámica, las Redes Neuronales deben tener una arquitectura recurrente para contemplar

la dependencia temporal, motivo por el cual se denomina Redes Neuronales Recurrentes o *Recurrent Neural Networks* (RNN) a los modelos apropiados bajo un contexto de series de tiempo.

Una RNN es un tipo especial de Red Neuronal creada para manejar datos secuenciales, como documentos de texto o series temporales, por lo que son básicamente Redes Neuronales con memoria. En su formulación general se trata de modelos con una estructura similar a la de una Red Neuronal Profunda donde es posible identificar neuronas de entrada, internas y de salida. Sin embargo, en una RNN la neurona tiene un estado oculto que almacena información sobre etapas anteriores, a la vez que no tiene una estructura de capas definida, sino que permite conexiones arbitrarias y recurrentes entre las neuronas. En este tipo de modelos la información circula tanto hacia adelante como hacia atrás durante el entrenamiento de la red gracias a que existen conexiones *forward* y *backward* entre las neuronas.

Gráfico 3. Arquitectura de una Red Neuronal Recurrente



Fuente: Jansen (2021)

Si consideramos una serie de tiempo $x = (x_0, x_1, x_2, \dots, x_t)$, inicialmente x_0 ingresa a la primera neurona recurrente, la cual brinda dos resultados: un estado oculto h_0 y un *output* y_0 . Luego x_1 es brindado como *input* en la segunda neurona recurrente, la cual calcula h_1 dado h_0 y x_1 y, posteriormente, calcula el *output* y_1 . De esta forma sucesiva, dicho proceso continúa durante toda la secuencia de x , a través del cual la RNN es capaz de almacenar información de períodos previos.

En estos modelos, el proceso de aprendizaje también se lleva a cabo ajustando los pesos de toda la Red Neuronal. El mecanismo de calibración que aplica una RNN es

similar al de una Red Neuronal no recurrente y se realiza mediante propagación hacia atrás a través del tiempo o *back-propagation through time* (BPTT). Como ya se mencionó anteriormente, cuando la red genera valores de predicción, también calcula el error de dicho pronóstico. Para minimizar el error ajustando las ponderaciones, primero la BPTT calcula las derivadas parciales (también conocidas como gradientes) del error con respecto a los pesos. Luego, la RNN recalibra los pesos, hacia arriba o hacia abajo, en función de las derivadas parciales.

En cuanto a su clasificación, las RNN generalmente se pueden clasificar en redes completamente recurrentes y parcialmente recurrentes, donde ambas clases de modelos pueden ser aproximadores universales³ para sistemas dinámicos. Por un lado, las redes completamente recurrentes permiten que cualquier neurona de la red exhiba conexiones hacia cualquier otra neurona. Sin embargo, estos modelos poseen problemas de estabilidad durante el entrenamiento y requieren algoritmos complejos y lentos. Por otro lado, las redes parcialmente recurrentes contienen retroalimentaciones internas hacia atrás, pero las interconexiones entre las neuronas son estrictamente hacia adelante; por lo que tienen una estructura menos complicada y permiten un entrenamiento más sencillo.

Generalmente, cuando se habla de redes neuronales parcialmente recurrentes, existe un grupo de neuronas especiales conocidas como neuronas de contexto, las cuales actúan como un *input* adicional para la capa oculta en el próximo período. Por lo tanto, existen dos tipos de neurona en la entrada de la red: las neuronas de entrada que reciben *inputs* desde el exterior y las neuronas de contexto que actúan como neuronas receptoras de las conexiones recurrentes provenientes de etapas previas que funcionan como una memoria de la red en las que se almacenan las activaciones de las iteraciones anteriores. Si se concatenan las activaciones de las neuronas de entrada y de contexto, una red parcialmente recurrente se puede ver como una red no recurrente multicapa. De esta manera, en las redes parcialmente recurrentes el cálculo de las activaciones de todas las neuronas se realiza como en una red multicapa sin recurrencias, es decir, desde la capa de entrada hacia la capa de salida, pasando por la capa oculta.

³ El Teorema de Aproximación Universal sostiene que una Red Neuronal con al menos una capa oculta y una cantidad suficiente de neuronas, puede aproximar cualquier función continua con cualquier grado de precisión.

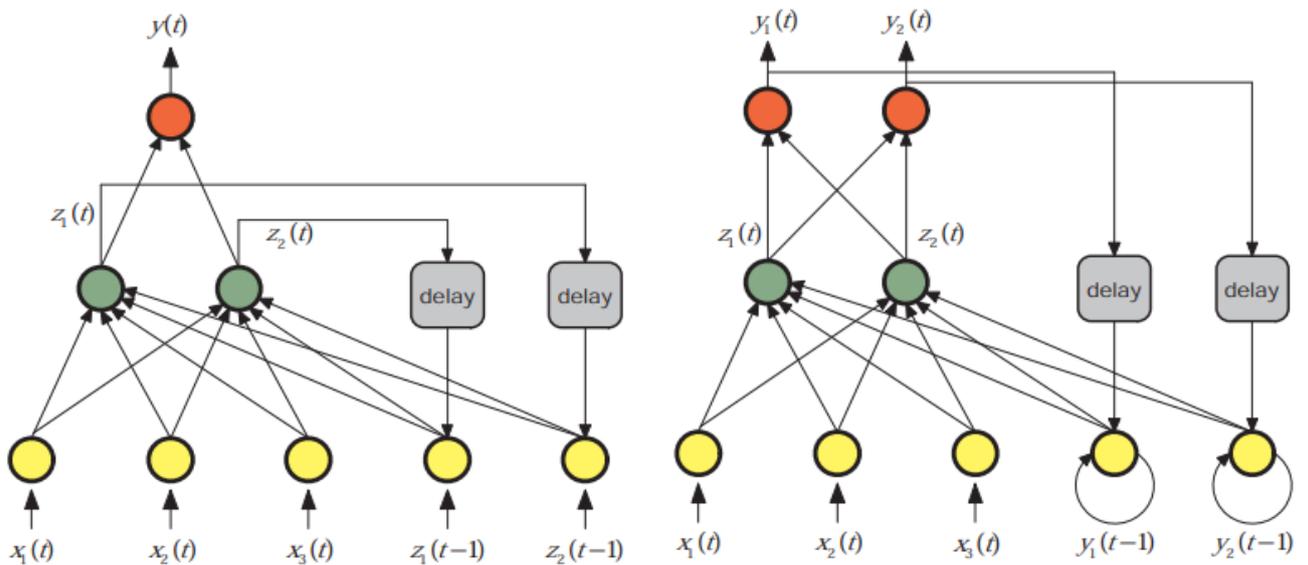
En este trabajo, voy a utilizar dos modelos de Redes Neuronales Recurrentes: la Red Neuronal de Jordan en representación de las redes neuronales parcialmente recurrentes y el modelo LSTM en representación de las redes neuronales completamente recurrentes.

3.5.1 Jordan

Las Redes Neuronales de Jordan y Elman son los modelos más utilizados dentro de las Redes Neuronales parcialmente recurrentes, los cuales son muy similares entre sí. En las redes de Elman los *inputs* de las neuronas de contexto provienen de los *outputs* de las neuronas de las capas ocultas. En cambio, en las redes de Jordan los *inputs* de las neuronas de contexto provienen de los *outputs* de las neuronas de las capas de salida de la red.

En los diagramas del Gráfico 4 se ilustran la arquitectura de ambos modelos. Los nodos amarillos representan las neuronas de la capa de entrada (x) y las de contexto (y o z según corresponda), los nodos verdes las neuronas de la(s) capa(s) oculta(s), los nodos rojos las neuronas de las capas de salida y los nodos grises representan los operadores de retardo, el cual indica que la señal es retardada en cierta unidad de tiempo.

Gráfico 4. Arquitectura de una Red Neuronal de Elman (izquierda) y Jordan (derecha)



Fuente: Drchal (2006)

En mayor detalle, Jordan propuso una arquitectura donde la capa de entrada está dividida en dos partes: el conjunto de *inputs* externos y la retroalimentación de la activación de la capa de salida a través de conexiones de valor fijo. Por lo tanto, esta red se caracteriza porque las neuronas de contexto reciben una copia de las neuronas de salida de la red y de ellas mismas. Además, las conexiones recurrentes de la capa de salida a las neuronas de contexto llevan un parámetro asociado, que toma un valor constante positivo y menor que 1. Posteriormente, Elman modifica esta arquitectura, considerando realimentaciones desde capas ocultas hacia la capa de contexto, pero sin considerar realimentaciones locales.

3.5.2 Long Short Term Memory

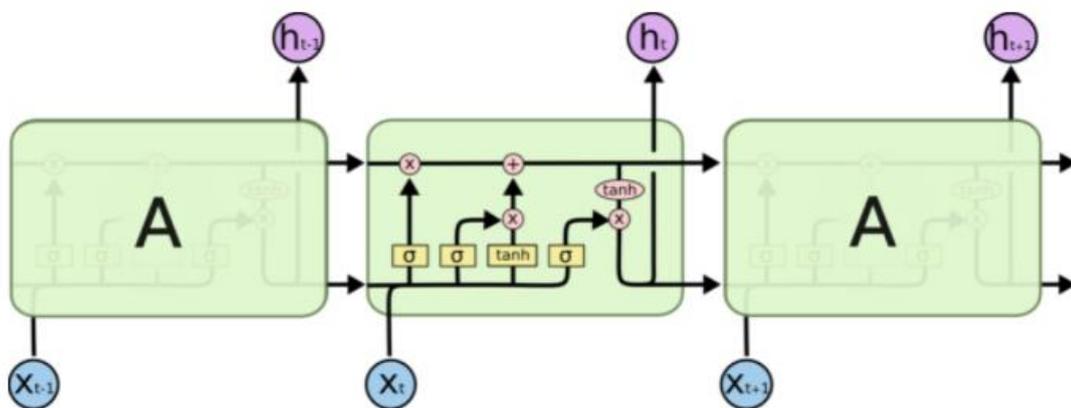
El modelo LSTM es una RNN capaz de aprender asociaciones de largo plazo sobre las series temporales. Esto lo hace generando gradientes de larga duración, los cuales resolvieron dos problemas latentes en la mayoría de las RNN: gradientes que se desvanecen o explotan. El problema del desvanecimiento del gradiente se refiere al hecho de que el gradiente converge rápidamente a cero a medida que itera hacia atrás en el tiempo. Al efectuar la propagación hacia atrás, si los ajustes de las ponderaciones sobre las capas anteriores son pequeños, los ajustes sobre la capa actual son aún menores y los gradientes se reducen exponencialmente. En este escenario, el aprendizaje es muy bajo ya que las ponderaciones apenas se corrigen debido a gradientes extremadamente pequeños o llanos, por lo que el modelo no puede mejorar las ponderaciones y reducir el error de predicción. Debido al problema del desvanecimiento del gradiente, la mayoría de las RNN no suelen poder modelar con precisión las dependencias de largo plazo de las series temporales y la memoria de la red se limita a las asociaciones de corto plazo. Por este motivo, el modelo LSTM busca mantener los gradientes lo suficientemente empinados para que la búsqueda del conjunto de ponderaciones que minimiza el error no se atasque. En cambio, el problema del gradiente explosivo se refiere al efecto opuesto, donde el gradiente se vuelve cada vez más grande y genera inestabilidad en el entrenamiento de la red.

La principal diferencia entre la arquitectura del modelo LSTM y la arquitectura de una red parcialmente recurrente radica en las operaciones que se efectúan dentro de las neuronas, las cuales permiten que el flujo de información esté regulado a través de

“puertas” que retienen o eliminan la información de manera selectiva. El modelo tiene 4 tipos de puertas, las cuales son funciones logísticas, que administran la información: la puerta de olvido f_t , la puerta de entrada i_t , la puerta de nuevos valores candidatos \tilde{C}_t y la puerta de salida o_t . El modelo decide “abrir” o “cerrar” estas puertas en función de los pesos que asigna cuando busca minimizar el error. Con el entrenamiento de la red, el modelo LSTM aprende qué información es relevante para reducir el error de predicción y abrirá o cerrará las puertas en función de las ponderaciones que asigne (con valores entre 0 y 1) a los *inputs* externos y provenientes de capas anteriores. En particular, si el modelo asigna un valor “elevado” cercano a 1 retendrá la información, mientras que si asigna un valor “pequeño” cercano a 0 descartará la información.

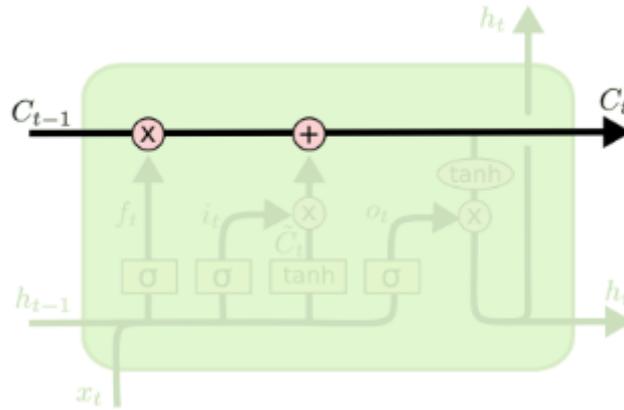
Sea n el número de neuronas en una capa del modelo LSTM, m el número de variables que funcionarán como *inputs* y $t = (1, \dots, T)$ el número de observaciones temporales. x_t es un vector de *inputs* de tamaño $(m \times 1)$ y h_t un vector de *outputs* de tamaño $(n \times 1)$. Sean $(W_f, W_i, W_{\tilde{c}}, W_o) \in \mathbb{R}^{n \times m}$ las matrices que contienen los pesos de las conexiones entre los *inputs* y las puertas de olvido, entrada, de valores candidatos y de salida respectivamente, $(U_f, U_i, U_{\tilde{c}}, U_o) \in \mathbb{R}^{n \times n}$ las matrices que contienen los pesos de las conexiones recurrentes de las puertas, y $(b_f, b_i, b_{\tilde{c}}, b_o) \in \mathbb{R}^{n \times 1}$ los vectores que contienen los sesgos de las puertas. Los pesos y sesgos se fijan inicialmente de forma aleatoria en un valor entre -1 y 1. A continuación, se presenta una serie de diagramas que ilustran la arquitectura del modelo LSTM en base a Olah (2015).

Gráfico 5. Arquitectura de una Red Neuronal Recurrente LSTM



Fuente: Olah (2015)

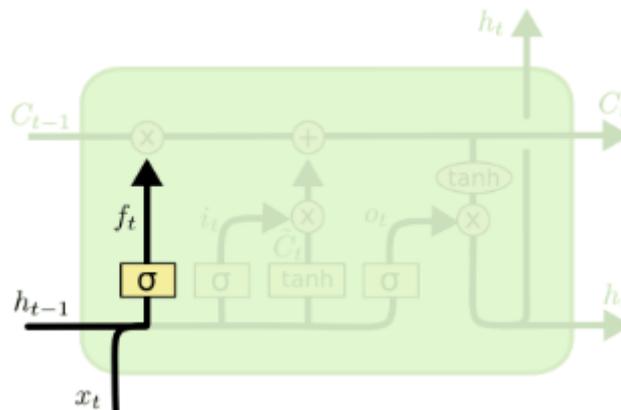
Gráfico 6. El concepto central de los modelos LSTM



Fuente: Olah (2015)

La clave de los modelos LSTM es el vector estado de las neuronas recurrentes C_t , el cual se ve representado a través de la línea horizontal que atraviesa la parte superior del Gráfico 6 y funciona como un medio que transfiere la información a lo largo de la red; por lo que se la puede pensar como la memoria del modelo. A medida que C_t avanza a lo largo de la red, la información se agrega o elimina de ella a través de las puertas. Estas puertas contienen activaciones sigmoides que generan números entre 0 y 1. Como indica la lógica, un valor igual 0 elimina la información, mientras que un valor igual a 1 guarda o permite que fluya toda la información.

Gráfico 7. La puerta de olvido



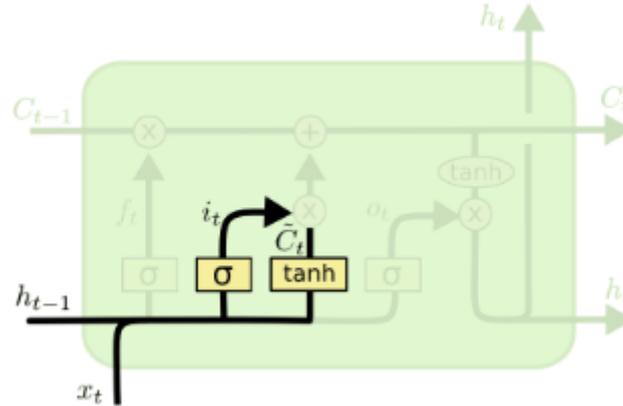
Fuente: Olah (2015)

$$f_t = \sigma (W_f x_t + U_f h_{t-1} + b_f) \quad (24)$$

El primer paso consiste en decidir qué información se guarda o se elimina de C_{t-1} . Esta decisión la toma la puerta de olvido (f_t), la cual analiza el resultado de la capa

anterior (h_{t-1}) y los *inputs* externos (x_t), utiliza una función sigmoidea y genera un número entre 0 y 1 para cada valor de C_{t-1} .

Gráfico 8. La puerta de entrada y la puerta de valores candidatos



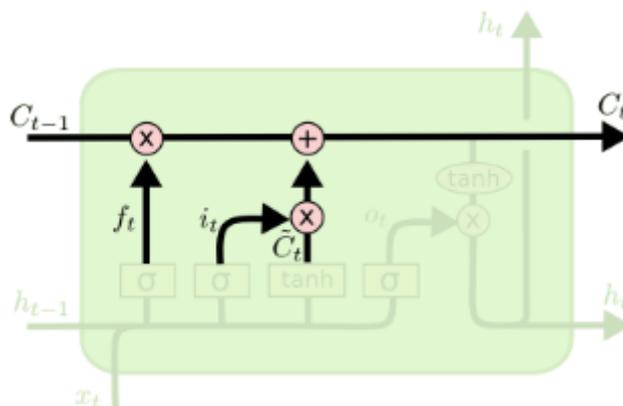
Fuente: Olah (2015)

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (25)$$

$$\tilde{C}_t = \tanh(W_{\tilde{C}} x_t + U_{\tilde{C}} h_{t-1} + b_{\tilde{C}}) \quad (26)$$

Una vez eliminada la información innecesaria, el siguiente paso consiste en decidir la nueva información que hay que incorporar a C_t . Esto tiene dos partes. Primero, la puerta de entrada i_t decide qué valores de C_t hay que actualizar con la nueva información. Para ello toma h_{t-1} y x_t como *inputs*, utiliza una función sigmoidea y genera un número entre 0 y 1 para cada valor de C_t . Luego, la puerta de la neurona \tilde{C}_t crea un vector de nuevos valores candidatos para actualizar C_t . Para ello toma h_{t-1} y x_t como *inputs*, utiliza una función \tanh y genera un número entre -1 y 1 para cada valor de C_t .

Gráfico 9. Actualización del vector estado de las neuronas recurrentes

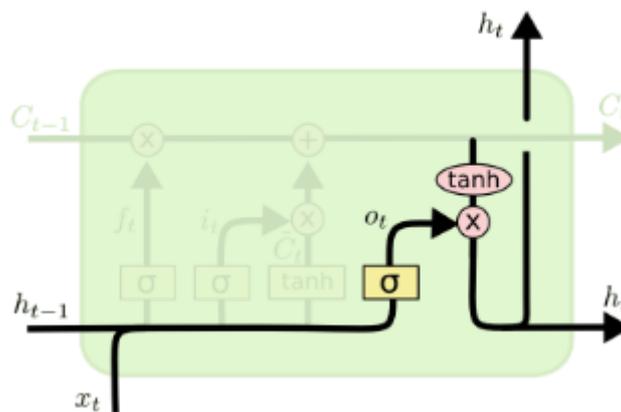


Fuente: Olah (2015)

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (27)$$

Una vez que i_t y \tilde{C}_t son calculados, se actualiza el estado de la neurona anterior C_{t-1} al nuevo estado de la neurona C_t combinando la información que se decidió eliminar y almacenar. Los pasos anteriores ya decidieron qué hacer, por lo que ahora solo se necesita llevarlo a cabo. En este sentido, se multiplica el estado anterior C_{t-1} por la puerta de olvido f_t , y se suma los nuevos valores candidatos \tilde{C}_t escalados por cuánto decidimos actualizar cada valor a través de la puerta de entrada i_t .

Gráfico 10. La puerta de salida



Fuente: Olah (2015)

$$o_t = \sigma (W_o x_t + U_o h_{t-1} + b_o) \quad (28)$$

$$h_t = o_t * \tanh(C_t) \quad (29)$$

Finalmente, la puerta de salida o_t decide cuál debe ser el resultado final de cierta etapa a través de dos funciones. Primero, la puerta de salida o_t decide qué parte de C_t va a ser un *output* tomando h_{t-1} y x_t como *inputs* y utilizando una función sigmoidea para generar un número entre 0 y 1. Luego, el próximo estado oculto h_t se basará en una versión filtrada de C_t a través de la función \tanh que multiplica a la puerta de salida o_t . De esta forma, el nuevo vector estado de las neuronas recurrentes C_t y el nuevo estado oculto h_t se transfieren al siguiente período de tiempo.

A modo de síntesis, la puerta de olvido f_t decide lo que es importante evitar de los pasos anteriores, la puerta de entrada i_t decide qué información es relevante actualizar y agregar, la puerta \tilde{C}_t genera nuevos valores candidatos para C_t y la puerta de salida o_t determina los resultados que se transfieren a la siguiente etapa..

Después de las capas ocultas, se puede especificar una capa de salida de p neuronas. Sea y un vector de tamaño $(px1)$ de *outputs*, definimos a $W_y \in \mathbb{R}^{pxn}$ como una matriz que contiene los pesos que conectan a la última capa oculta con la capa de salida y a b_y como un vector de tamaño $(px1)$ que contiene los sesgos. Entonces el vector de *outputs* y devuelto por la capa de salida está dado por $Y = g(W_y h_t + b_y)$, donde $g(\cdot)$ es la función de activación de la capa de salida.

4. Base de datos

4.1 Indicadores

Para los modelos multivariados, en este trabajo se utilizan 32 indicadores económicos adicionales (sin incluir al EMAE s.e.) de diversa índole con frecuencia mensual desde el año 2005 hasta el año 2021, los cuales se encuentran detallados en la Tabla 2 y constituyen el conjunto de potenciales predictores del EMAE s.e. Estos indicadores contienen datos sectoriales, de comercio exterior, fiscales, monetarios, cambiarios, financieros y de confianza. Dentro de los indicadores sectoriales se encuentran datos agropecuarios, agro-industriales, industriales, de petróleo & gas, de construcción, de transporte y de actividades inmobiliarias. La base de datos también contempla una variable binaria para capturar la irrupción de la pandemia en marzo y abril de 2020 (adopta un valor igual a 1 en dichos meses e igual a 0 en los restantes). Todos los indicadores se encuentran desestacionalizados a excepción de la variable binaria. Además, los datos monetarios y cambiarios constituyen promedios mensuales ya que las series originales tienen frecuencia diaria, mientras que los datos fiscales y monetarios también son deflactados por el Índice de Precios al Consumidor para que se encuentren expresados en términos reales.

Tabla 2. Indicadores económicos

Sector	Variable	Unidad de medida	Fuente
Agro	Producción de Soja	Millones de toneladas	Ministerio de Economía
	Producción de Maíz	Millones de toneladas	Ministerio de Economía
	Producción de Trigo	Millones de toneladas	Ministerio de Economía
	Producción de leche	Millones de litros	Ministerio de Agricultura
Agro-industria	Molienda de soja	Toneladas	Ministerio de Agricultura
	Molienda de girasol	Toneladas	Ministerio de Agricultura
Industria	Patentamientos de automóviles	Unidades	SIOMAA
	Producción de automóviles	Unidades	ACARA
	Ventas de automóviles a concesionarios	Unidades	ACARA
	Exportaciones de automóviles	Unidades	ACARA
	Despachos de cemento	Miles de toneladas	AFCP
	Producción de hierro primario	Miles de toneladas	Cámara Argentina del Acero
	Producción de acero crudo	Miles de toneladas	Cámara Argentina del Acero
	Demanda de electricidad grandes usuarios	GWh	CAMMESA
Petróleo & Gas	Producción de petróleo	Miles de metros cúbicos	Ministerio de Economía
	Producción de gas	Millones de metros cúbicos	Ministerio de Economía
Construcción	Índice Construya	Jun-02 = 100	Grupo Construya
Transporte	Transporte de pasajeros de red ferroviaria AMBA	Pasajeros	Ministerio de Transporte
	Transporte de trenes de carga	Millones Tn * Kilómetro	Ministerio de Transporte
	Vuelos aéreos de cabotaje	Vuelos	ANAC
Actividades inmobiliarias	Escrituras en CABA	Cantidad de escrituras	Colegio de Escribanos CABA
	Compraventa de inmuebles en PBA	Cantidad de actos	Colegio de Escribanos PBA
Comercio exterior	Volumen de exportaciones de bienes	Índice 2004 = 100	INDEC
	Volumen de importaciones de bienes	Índice 2004 = 100	INDEC
Fiscal	Recaudación tributaria ligada a la actividad	\$mn ctes.	Ministerio de Economía
Monetario	Depósitos del sector privado en pesos	\$mn ctes.	BCRA
	Préstamos al sector privado en pesos	\$mn ctes.	BCRA
	M2 privado	\$mn ctes.	BCRA
Cambiario	Tipo de cambio real	17-12-2015 = 100	BCRA
Financiero	Índice Merval	Volumen negociado	Reuters
Confianza	Confianza al consumidor	Índice entre 0 y 100	UTDT
Otros	Variable binaria – pandemia	Valores 0 o 1	-

Fuente: elaboración propia

En mayor detalle, vale la pena realizar algunos comentarios adicionales. La producción de cereales (soja, trigo, maíz) se trata de una actividad de ciclo anual. El INDEC distribuye temporalmente la producción agrícola en función de coeficientes

mensuales de avance de las distintas labores culturales y se sigue mediante los pronósticos de área sembrada, cosechada y volúmenes de producción elaborados por fuentes oficiales y privadas. En este caso, la serie mensual se obtiene utilizando las ponderaciones implícitas en las publicaciones del EMAE que realizó el INDEC entre 2017 y 2021. Como los pesos de cada mes en el año son similares entre los diferentes años para los cuales hay información, la serie mensual se reconstruye hacia atrás utilizando un promedio de las ponderaciones de cada mes entre los distintos años. Además, la recaudación tributaria solo contiene los rubros que se asocian a la actividad económica, los cuales incluyen el impuesto al valor agregado, el impuesto a los créditos y débitos de cuenta corriente y el impuesto a los combustibles.

Por último, vale la pena mencionar que una serie de indicadores sectoriales fueron excluidos de la base de datos ante la falta de información:

1. Producción de carne bovina: la serie con frecuencia mensual comienza en 2019.
2. Producción de carne porcina: la serie con frecuencia mensual comienza en 2017.
3. Producción de carne aviar: la serie con frecuencia mensual comienza en 2016.
4. Comercio minorista relevado por la Confederación Argentina de la Mediana Empresa: la serie comienza en el 2008.

4.2 Transformación de los datos

Para obtener series desestacionalizadas se identifican y remueven las fluctuaciones estacionales y los efectos del calendario. En particular, el ajuste estacional que efectúa el INDEC se realiza por el método indirecto, es decir, el EMAE s.e. se obtiene como agregación de las desestacionalizaciones efectuadas a cada uno de los sectores productivos. Según el INDEC, dicho método tiene tres ventajas con respecto al método directo: (i) satisface criterios deseables de suavidad, (ii) permite explicar los movimientos del agregado a partir de los movimientos de sus componentes y (iii) si se construye un indicador desestacionalizado trimestral a partir del EMAE s.e., este tiene discrepancias menores con respecto al PBI s.e. Según el INDEC, todos los sectores productivos que componen el PBI a precios básicos (excepto la minería, la actividad financiera, los servicios de hogares privados que contratan servicio doméstico y los impuestos a los débitos y créditos) tienen estacionalidad identificable en términos estadísticos.

En este trabajo se utiliza directamente el EMAE s.e. que publica el INDEC, mientras que el resto de los indicadores económicos son desestacionalizados con el programa Demetra. Los resultados de la desestacionalización son “buenos” para todas las variables a excepción de la producción de maíz que es “incierto”. Un resultado “bueno” indica que el resultado es bueno desde el punto de vista de la calidad del ajuste estacional ya que ninguno de los indicadores cualitativos es “severo”. En cambio, tal como su nombre lo indicia, un resultado “incierto” significa que la calidad del ajuste estacional es incierta; mientras un resultado “severo” (el cual no es alcanzado por ninguno de los indicadores seleccionados en este trabajo) muestra que no existen errores lógicos en los resultados, pero que estos no deben ser aceptados por motivos de calidad del ajuste estacional ya que el valor de al menos uno de los indicadores cualitativos es “severo”.

En relación a los distintos métodos de estimación, se realizan algunas transformaciones adicionales:

1. Modelos VAR y FAVAR: se trabaja con las diferencias del logaritmo del EMAE s.e. y de las variables explicativas que se utilizan en cada modelo para que sean estacionarias.
2. Modelo de Redes Neuronales Recurrentes (Jordan y LSTM): se trabaja con la base de datos estandarizada (a cada variable se le resta la media y se la divide por el desvío) para acelerar el proceso de convergencia en la estimación. En esta línea, Zhang & Qi (2005) encontraron que las Redes Neuronales no son capaces de capturar la estacionalidad de las series temporales, por lo que desestacionalizar las series puede mejorar el desempeño predictivo del modelo. Además, Balkin & Keith (2000) sostienen que diferenciar las series para una Red Neuronal puede ser innecesario para proyectar, pero que ciertas transformaciones como escalar las variables puede ser beneficioso.

En todos estos casos las estimaciones son posteriormente re-escaladas para obtener los resultados finales.

5. Estimaciones y resultados

El conjunto de entrenamiento inicial (*Train*) para todos los modelos comprende a los datos entre enero de 2005 y diciembre de 2016 (144 observaciones temporales). Por lo tanto, el conjunto de validación o testeo (*Test*) abarca al período entre enero de 2017 y diciembre de 2021 (60 observaciones temporales). Sin embargo, es importante aclarar que el conjunto de entrenamiento se actualiza mes a mes con el objetivo de realizar las proyecciones a un mes vista con la última información disponible. A modo ilustrativo, para proyectar el EMAE s.e. de julio de 2021, el conjunto *Train* comprende a los datos entre enero de 2005 y junio de 2021.

Evaluar el desempeño predictivo de un modelo es una de las tareas más importantes en el proceso de generación de pronósticos. En este sentido, la raíz del error cuadrático medio o *root mean square error* (RMSE) y el error porcentual absoluto medio o *mean absolute percentage error* (MAPE) serán las métricas utilizadas para evaluar el desempeño relativo de cada método de estimación.

$$RMSE = \sqrt{\frac{1}{T} \sum_{t=1}^T (\hat{y}_t - y_t)^2} \quad (30)$$

$$MAPE = \frac{1}{T} \sum_{t=1}^T \left| \frac{\hat{y}_t - y_t}{y_t} \right| * 100 \quad (31)$$

El RMSE es una función de pérdida que depende de la escala de las variables y es más sensible a *outliers*; mientras que el MAPE es independiente de la magnitud de las variables, aunque no puede ser utilizado cuando la variable en cuestión es igual a cero y tiene una distribución sesgada cuando se acerca a cero. Sin embargo, en este trabajo, la variable de interés (EMAE s.e.) nunca se acerca a cero en todo el período de análisis.

Además, en el caso de los modelos de *Machine Learning* (SSA, Jordan y LSTM) se buscará la combinación de hiper-parámetros que arroje el menor error de predicción. Para lograr esto, los métodos de aprendizaje supervisado son entrenados en el conjunto *Train* y se evalúan diferentes combinaciones de los hiper-parámetros de cada modelo en

el conjunto *Test*, donde se selecciona aquella combinación que arroje el menor RMSE. Una vez seleccionado la “mejor versión” de cada modelo de *Machine Learning*, se evalúa su desempeño con respecto a los modelos econométricos para el conjunto *Test* y para toda la muestra, donde en este último caso también se realiza una proyección para enero de 2022. En este marco, se presentan dos gráficos del EMAE s.e. correspondiente a toda la muestra (enero de 2005 - diciembre de 2021) y al conjunto *Test* (enero de 2017 - diciembre de 2021).

Gráfico 11. Evolución del EMAE s.e.



Fuente: elaboración propia en base a INDEC

5.1 Estimaciones para el conjunto *Test*

5.1.1 ARIMA

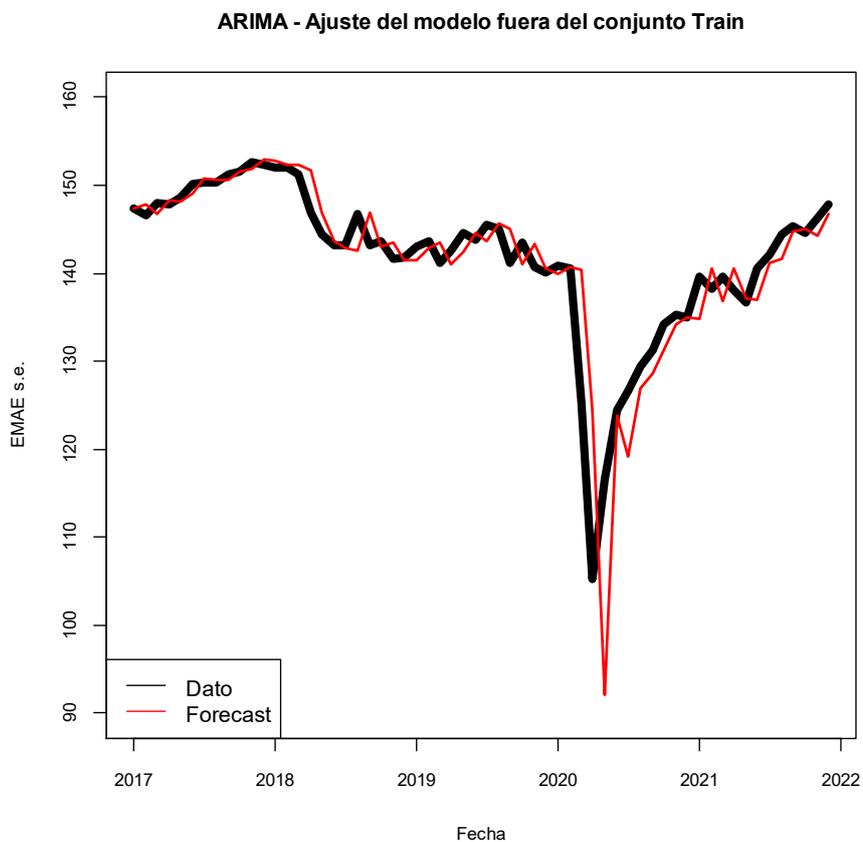
Para seleccionar el modelo ARIMA más apropiado se utiliza la función *auto.arima* de la librería *forecast* en *R*. Esta función usa una variante del algoritmo Hyndman-Khandakar⁴, la cual combina pruebas de raíces unitarias, una minimización del criterio de información de Akaike corregido por muestra pequeña (AICc) y una estimación por máxima verosimilitud para obtener el mejor modelo ARIMA. La función *auto.arima* es una herramienta útil que proporciona una opción rápida para construir modelos y pronósticos con series temporales, debido a que evalúa entre todos los posibles modelos e identifica de forma automática al mejor de ellos considerando diversos criterios. Como el conjunto de entrenamiento se actualiza mes a mes, en cada proyección a un mes vista la estimación se puede realizar con un modelo ARIMA(p,d,q) determinado. A modo ilustrativo, si se considera a toda la muestra como período de

⁴ Hyndman, R. J.; Khandakar, Y. (2008). *Automatic time series forecasting: The forecast package for R*. Journal of Statistical Software. Vol. 27, No. 3. Disponible en: <https://www.jstatsoft.org/article/view/v027i03>

entrenamiento, el modelo ARIMA(0,1,2) es el más apropiado para proyectar el EMAE s.e. en enero de 2022 (más detalles en la sección 5.2).

El modelo ARIMA brindó un RMSE y un MAPE de 4,92 y 1,95 respectivamente para el conjunto *Test*. A continuación, se presenta un gráfico con las proyecciones a un mes vista del EMAE s.e. entre enero de 2017 y diciembre de 2021 para el modelo ARIMA.

Gráfico 12. Ajuste del modelo ARIMA en el conjunto *Test*



Fuente: elaboración propia

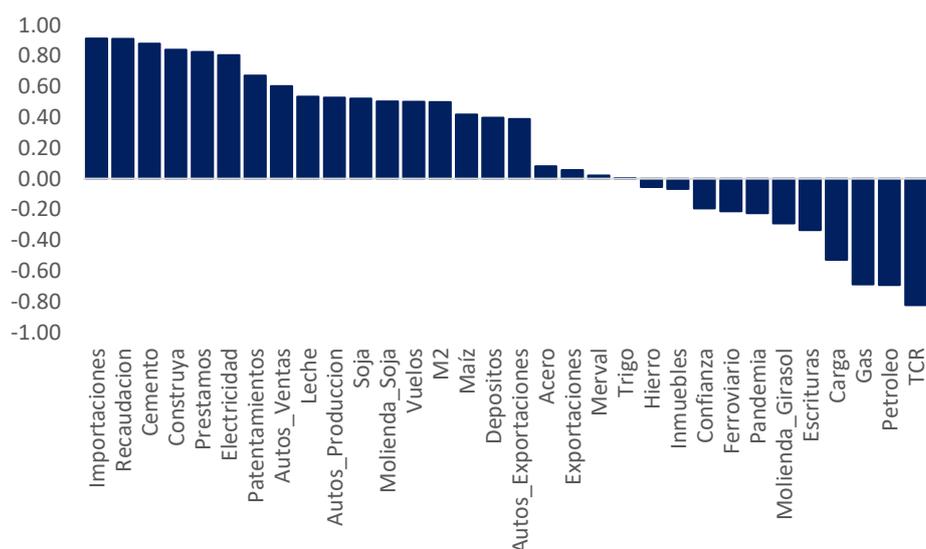
5.1.2 VAR

Para seleccionar el modelo VAR(p) más apropiado se utiliza la función *VARselect* de la librería *vars* en R. Esta función estima una serie de modelos VAR para una cantidad determinada de rezagos y devuelve diversos criterios de información y el *Final Prediction Error* (FPE) para cada uno de ellos. En particular, los posibles modelos candidatos incluyen una constante y contempla un máximo de 8 rezagos posibles debido a que lo

sucedido hace 9 meses difícilmente tenga una incidencia sobre la evolución actual de la actividad económica sin estacionalidad. Según los criterios de Akaike (AIC), de Schwarz (BIC), de Hannan-Quinn (HC) y el FPE; el VAR óptimo para el conjunto *Train* inicial es de orden 1. Por lo tanto, se procede a estimar el EMAE s.e. con un VAR(1).

Vale la pena mencionar que, con el objetivo de no estimar una gran cantidad de parámetros, para este modelo solo se incluye a las variables que exhibieron mayor correlación con el EMAE s.e. entre 2005 y 2021, tanto de manera positiva como negativa. Por el lado positivo, se incorpora al volumen de las importaciones de bienes (0,91), la recaudación tributaria ligada a la actividad (0,91), los despachos de cemento (0,88), el índice Construya (0,84), los préstamos al sector privado (0,83) y la demanda de electricidad de grandes usuarios (0,80). En cambio, por el lado negativo, se incluye al tipo de cambio real (-0,82), la producción de petróleo (-0,69) y la producción de gas (-0,69). Por lo tanto, al pasar de 33 a 10 variables (contemplando al EMAE s.e.), se reduce notoriamente la dimensionalidad de los datos. Además, también se efectúa un test de *Dickey-Fuller* para verificar que todas las series (las cuales están expresadas en diferencias del logaritmo en este modelo) sean estacionarias.

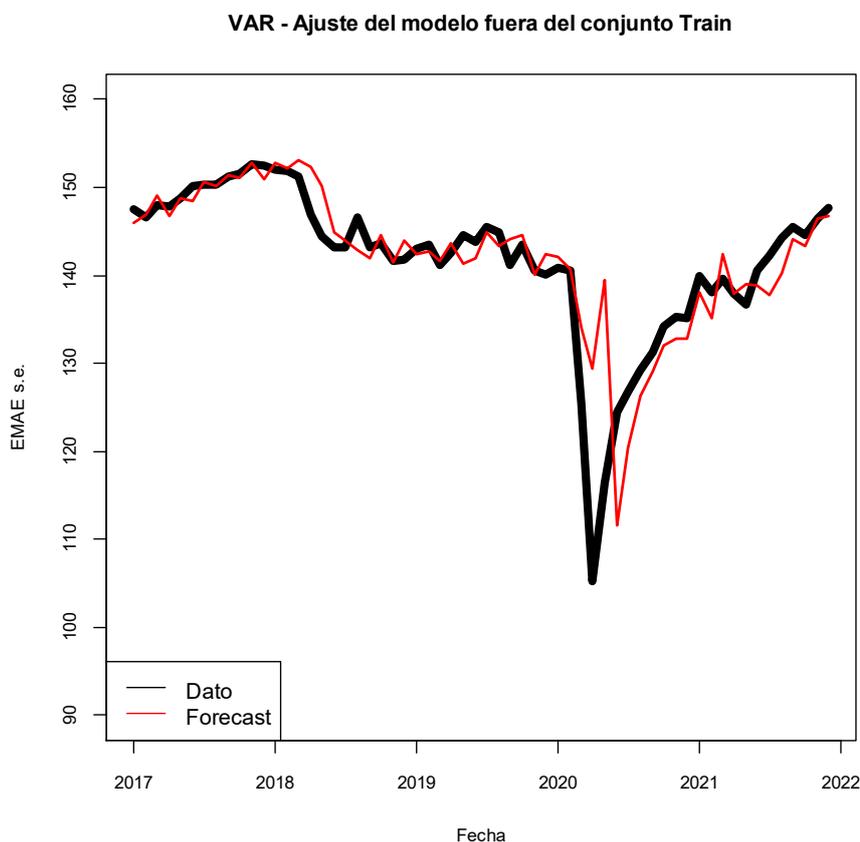
Gráfico 13. Coeficiente de correlación entre el EMAE s.e. y los diversos indicadores económicos entre 2005 y 2021



Fuente: elaboración propia en base a INDEC

El modelo VAR brindó un RMSE y un MAPE de 5,21 y 2,10 respectivamente para el conjunto *Test*. A continuación, se presenta un gráfico con las proyecciones a un mes vista del EMAE s.e. entre enero de 2017 y diciembre de 2021 para el modelo VAR.

Gráfico 14. Ajuste del modelo VAR en el conjunto *Test*



Fuente: elaboración propia

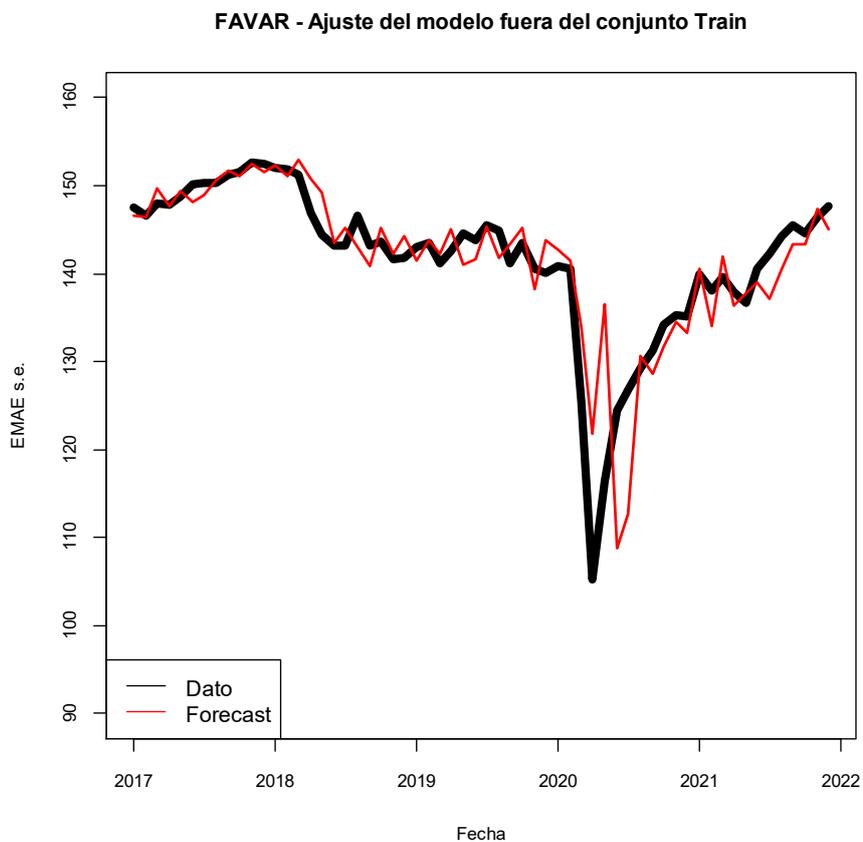
5.1.3 FAVAR

Para el modelo FAVAR, se excluye la producción de automóviles y la compraventa de inmuebles de la Provincia de Buenos Aires debido a que las series desestacionalizadas de estas variables adoptan valores negativos durante la irrupción de la pandemia, por lo que no se puede aplicar el logaritmo a dichas series para hacerlas estacionarias. Una vez hecho esto, al igual que en el modelo VAR, se efectúa un test de *Dickey-Fuller* para verificar que todas las series, las cuales también están expresadas en diferencias del logaritmo, sean estacionarias. Posteriormente, tras calcular los componentes principales de los datos, se estima un VAR con los primeros 5 componentes. En este caso, los criterios AIC, BIC, HC y el FPE indican que el VAR óptimo es de orden

4 para el conjunto *Train* inicial, por lo que se procede a realizar la estimación del EMAE s.e. con un VAR(4).

El modelo FAVAR brindó un RMSE y un MAPE de 4,91 y 2,18 respectivamente para el conjunto *Test*. A continuación, se presenta un gráfico con las proyecciones a un mes vista del EMAE s.e. entre enero de 2017 y diciembre de 2021 para el modelo FAVAR.

Gráfico 15. Ajuste del modelo FAVAR en el conjunto *Test*



Fuente: elaboración propia

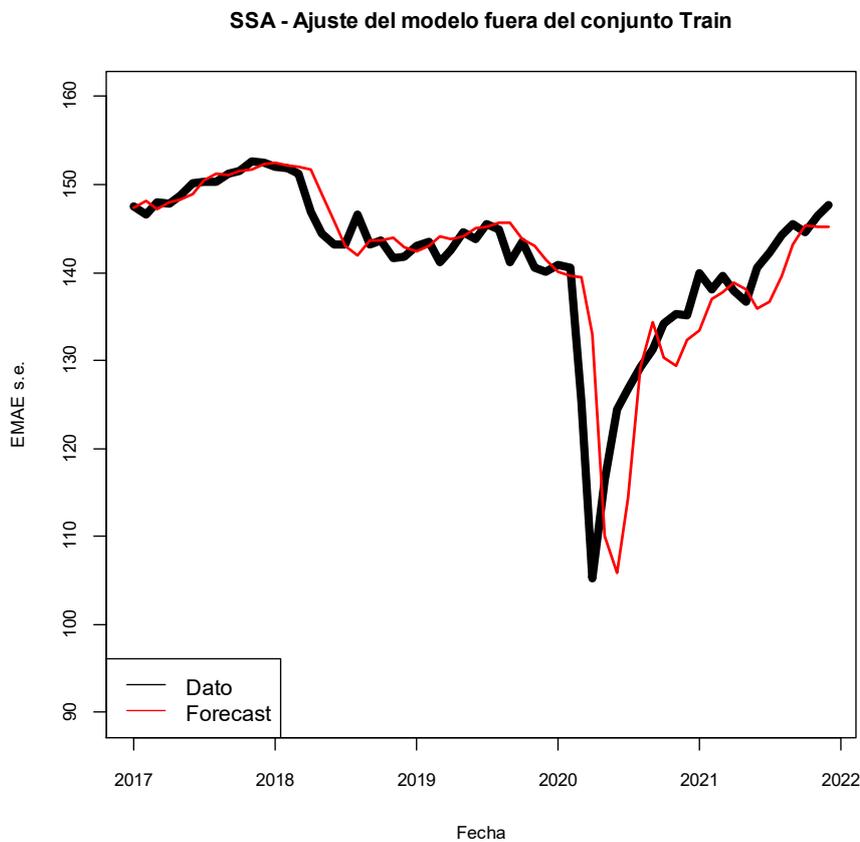
5.1.4 Análisis Espectral Singular

Para estimar el modelo SSA se utiliza la librería *ASSA* en *R*. Tal como se puede ver en las Tablas 1 y 1bis del Anexo, la grilla de valores candidatos para el hiper-parámetro *L* es (50, 70, 90, 95, 100, 105, 110, 115, 120, 125, 130), mientras que para el hiper-parámetro *M* es (2, 4, 6, 8, 10, 11, 12, 14, 16). La combinación de hiper-parámetros que arrojó el menor RMSE para el conjunto *Test* fue la siguiente: $L = 120$; $M = 11$.

Por un lado, el valor del hiper-parámetro L es lo suficientemente “elevado” para lograr una adecuada separación de los componentes, aunque se puede observar que se ubica por encima del valor óptimo que brindan algunas simulaciones mencionadas en la sección 3.4 ($0,33T-0,40T$) al representar casi el 60% de las observaciones temporales de todo el período de análisis. Además, vale la pena mencionar que, si bien el valor óptimo del hiper-parámetro L es un múltiplo de 12, esto no sería necesario en este trabajo ya que la variable de interés se encuentra en términos desestacionalizados. Por otro lado, el valor del hiper-parámetro M también es “elevado”, lo que resulta esperable dado que valores altos de M se asocian con proyecciones de corto plazo, tal como se comentó en el apartado 3.4.

Con $L = 120$ y $M = 11$, el modelo SSA brindó un RMSE y un MAPE de 5,53 y 2,26 respectivamente para el conjunto *Test*. A continuación, se presenta un gráfico con las proyecciones a un mes vista del EMAE s.e. entre enero de 2017 y diciembre de 2021 para el modelo SSA.

Gráfico 16. Ajuste del modelo SSA en el conjunto *Test*



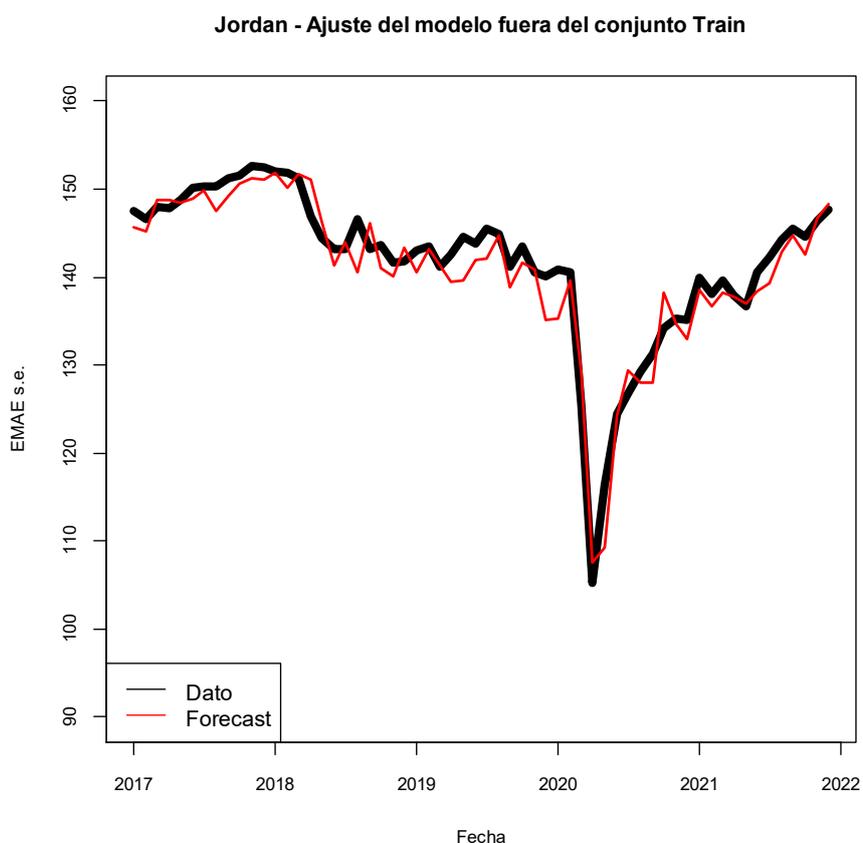
Fuente: elaboración propia

5.1.5 Jordan

Para estimar el modelo de Jordan se utilizan las librerías *Rcpp*, *RSNNS*, *zoo*, *xts*, *TTR*, *quantmod* y *tensorflow* en *R*. Tal como se puede ver en la Tabla 2 del Anexo, la grilla de valores candidatos para el hiper-parámetro k es (1, 2, 3, 4, 5, 7), mientras que para el hiper-parámetro λ es (0,1, 0,5). La combinación de hiper-parámetros que arrojó el menor RMSE fue la siguiente: $k = 3$; $\lambda = 0,1$, donde k representa la cantidad de neuronas en cada capa oculta y λ representa la tasa de aprendizaje. En este punto, vale la pena recordar que las redes más extensas o profundas permiten capturar patrones complejos de las series temporales, con un mayor riesgo de sobre-parametrización o *over-fitting* en los datos (menos sesgo y más varianza). En cambio, las redes más simples capturan patrones simples en los datos, con un mayor riesgo de sub-parametrización o de *under-fitting* en los datos (más sesgo y menos varianza). Asimismo, una tasa de aprendizaje elevada que se acerca a 1 puede acelerar el proceso de entrenamiento de la red, pero también evitar que la red se asiente en un mínimo del error total. En cambio, una baja tasa de aprendizaje permite un mejor ajuste del modelo a costa de una optimización más lenta.

Con $k = 3$; $\lambda = 0,1$, el modelo Jordan brindó un RMSE y un MAPE de 2,47 y 1,38 respectivamente para el conjunto *Test*. A continuación, se presenta un gráfico con las proyecciones a un mes vista del EMAE s.e. entre enero de 2017 y diciembre de 2021 para el modelo de Jordan.

Gráfico 17. Ajuste del modelo de Jordan en el conjunto *Test*



Fuente: elaboración propia

5.1.6 Long-Short Term Memory

El modelo LSTM puede ser estimado bajo un enfoque univariado y multivariado. Además, dado que el modelo LSTM contiene una memoria de largo plazo como se describió en la sección 3.5.2, es posible realizar las estimaciones sin actualizar el conjunto *Train* inicial. Por lo tanto, en este trabajo el modelo LSTM tiene cuatro variantes según la cantidad de variables y la actualización del conjunto *Train* inicial: LSTM univariado, LSTM multivariado, LSTM univariado *rolling* y LSTM multivariado *rolling*.

Para estimar el modelo LSTM se utilizan las librerías *keras* y *tensorflow* en *R*. La librería *keras* requiere que la información este organizada en matrices de tres dimensiones para la primera capa (*samples*, *timesteps* y *features*), tanto para la variable de interés como para las variables explicativas. *Samples* indica el número de observaciones temporales; *Timesteps* el número de rezagos incluidos en el conjunto *Train* y *Test* o, en otras palabras, cuántas unidades atrás en el tiempo queremos que vea la red;

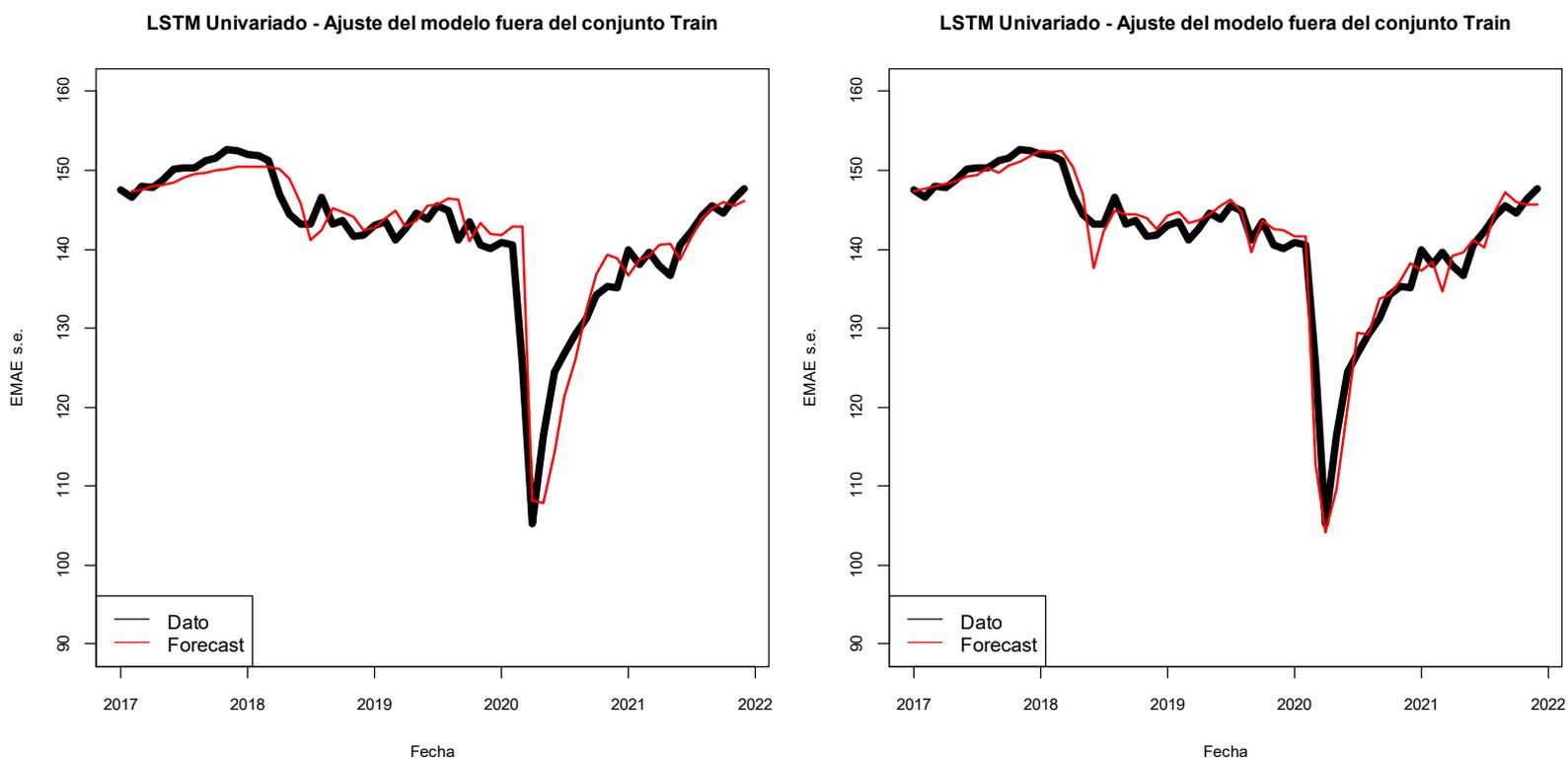
y *Features* el número de variables explicativas. Por lo tanto, a modo ilustrativo, si consideramos todo el período de análisis entre 2005 y 2021 la dimensión de las matrices sería la siguiente: [204, 1, 1] en el caso univariado o [204, 1, 33] en el caso multivariado.

Además, el algoritmo tiene tres etapas: la definición, la compilación y el ajuste del modelo. A la hora de definir el modelo, se establecen la cantidad de neuronas, la cantidad de capas y la tasa de aprendizaje. A la hora de compilar el modelo, la función de pérdida es el error cuadrático medio, la estimación adaptativa de momentos (ADAM) es el algoritmo de optimización y la precisión (*accuracy*) es la métrica para evaluar el desempeño del modelo. Por último, a la hora de ajustar el modelo, se determinan el número de *epochs*, concepto que hace referencia a cuántas veces el conjunto de datos de entrenamiento se envía a la Red Neuronal para entrenarla. Generalmente, a mayor *epochs*, el desempeño del modelo mejora, aunque puede generar *over-fitting* en los datos.

De lo anterior, se deduce que los modelos LSTM tienen una arquitectura compleja y una gran cantidad de hiper-parámetros. En este trabajo se optimizan tres de ellos: *neuronas*, *epochs*, *drop*. El hiper-parámetro *neuronas* representa la cantidad de neuronas en cada capa oculta, el hiper-parámetro *epochs* representa la cantidad de iteraciones y el hiper-parámetro *drop* representa la tasa de aprendizaje. En cambio, hay otros hiper-parámetros que permanecen fijos: todos los modelos tienen tres capas (una de entrada, una oculta y una de salida), el *batch size* es igual a 1, el algoritmo de optimización es ADAM y la función de activación es una *tanh*.

Tal como se puede ver en la Tabla 3 del Anexo, la grilla de valores candidatos para el hiper-parámetro *neuronas* es (10, 20, 50, 100, 200, 300), para el hiper-parámetro *epochs* es (10, 20, 50, 100, 150, 200, 250) y para el hiper-parámetro *drop* es (0,01, 0,1, 0,5). La combinación de hiper-parámetros que arrojó el menor RMSE para el modelo LSTM univariado fue la siguiente: *neuronas* = 10; *epochs* = 100; *drop* = 0,01. El modelo LSTM univariado brindó un RMSE y un MAPE de 3,65 y 1,92 respectivamente para el conjunto *Test*. Además, utilizando la misma combinación de hiper-parámetros, el modelo LSTM univariado *rolling* brindó un RMSE y un MAPE de 2,60 y 1,35 respectivamente para el conjunto *Test*. A continuación, se presenta un gráfico con las proyecciones a un mes vista del EMAE s.e. entre enero de 2017 y diciembre de 2021 para el modelo LSTM univariado y para el modelo LSTM univariado *rolling*.

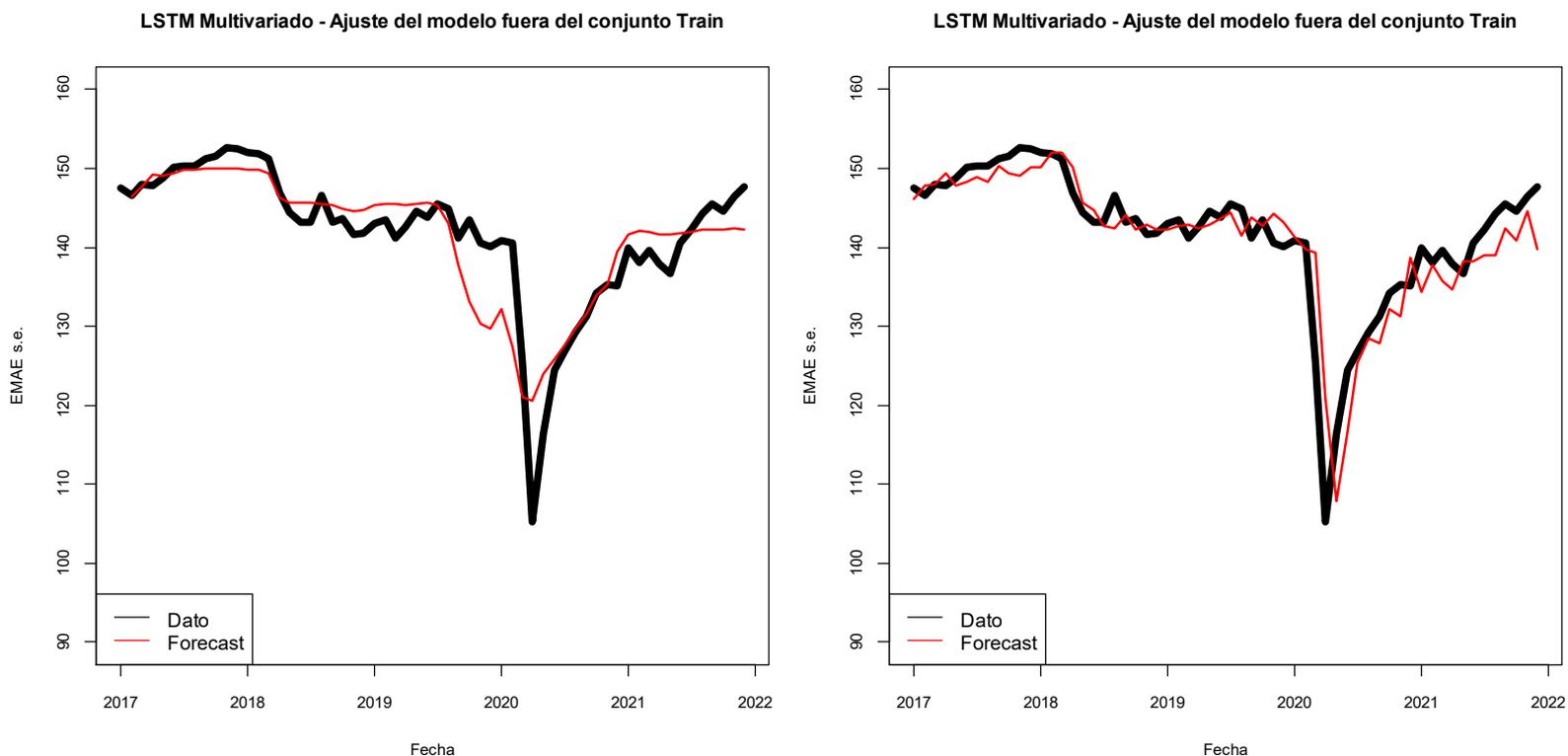
Gráfico 18. Ajuste del modelo de LSTM univariado (izquierda) y LSTM univariado *rolling* (derecha) en el conjunto *Test*



Fuente: elaboración propia

Tal como se puede ver en la Tabla 4 del Anexo, se utiliza la misma grilla de valores candidatos que en el enfoque univariado. En esta oportunidad, la combinación de hiper-parámetros que arrojó el menor RMSE para el modelo LSTM multivariado fue la siguiente: *neuronas* = 200; *epochs* = 100; *drop* = 0,1. El modelo LSTM multivariado brindó un RMSE y un MAPE de 4,43 y 2,37 respectivamente para el conjunto *Test*. Además, utilizando la misma combinación de hiper-parámetros, el modelo LSTM multivariado *rolling* brindó un RMSE y un MAPE de 3,97 y 2,13 respectivamente para el conjunto *Test*. A continuación, se presenta un gráfico con las proyecciones a un mes vista del EMAE s.e. entre enero de 2017 y diciembre de 2021 para el modelo LSTM multivariado y para el modelo LSTM multivariado *rolling*.

Gráfico 19. Ajuste del modelo de LSTM multivariado (izquierda) y LSTM multivariado *rolling* (derecha) en el conjunto *Test*



Fuente: elaboración propia

Con todo, los modelos de *Machine Learning* compiten con los modelos econométricos más tradicionales de series de tiempo para proyectar el EMAE s.e. De hecho, al analizar el RMSE y el MAPE de forma conjunta para todos los modelos, se puede verificar en la Tabla 3 que el podio está integrado por tres modelos de aprendizaje supervisado (Jordan, LSTM Univariado *rolling* y LSTM Univariado). Estos resultados se encuentran en línea con la mayoría de los hallazgos mencionados en la sección 2. Además, en el caso de los modelos LSTM, actualizar el conjunto *Train* es una estrategia dominante tanto para el enfoque univariado como multivariado ya que el RMSE y el MAPE disminuyen en ambos casos.

Tabla 3. RMSE y MAPE de los distintos modelos para el conjunto *Test*

	Test: 2017 -2021			
	RMSE	Ranking	MAPE	Ranking
ARIMA	4.92	7	1.95	4
VAR	5.21	8	2.10	5
FAVAR	4.91	6	2.18	7
SSA	5.53	9	2.26	8
Jordan	2.47	1	1.38	2
LSTM Univariado	3.65	3	1.92	3
LSTM Univariado <i>Rolling</i>	2.60	2	1.35	1
LSTM Multivariado	4.43	5	2.37	9
LSTM Multivariado <i>Rolling</i>	3.97	4	2.13	6

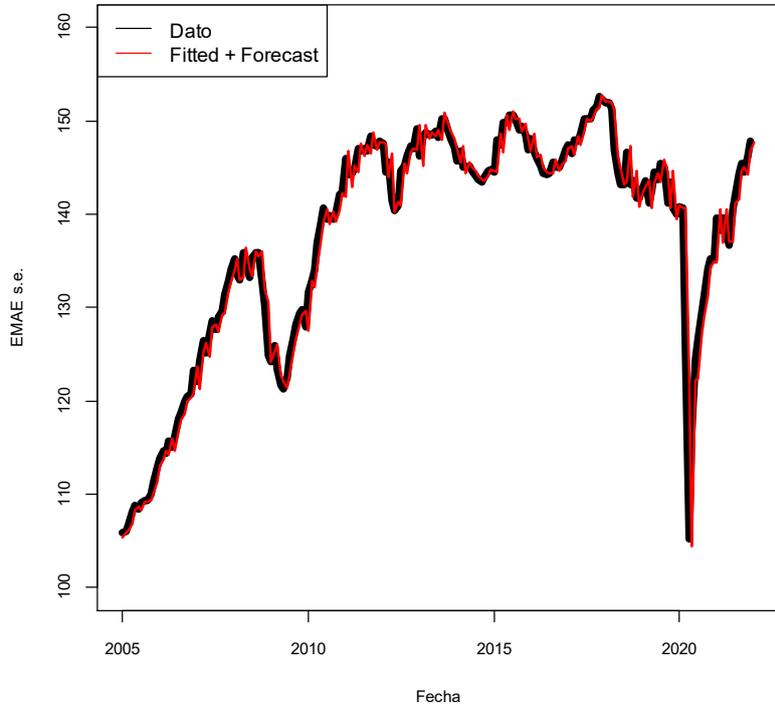
Fuente: elaboración propia

5.2 Estimaciones para toda la muestra y proyección de enero de 2022

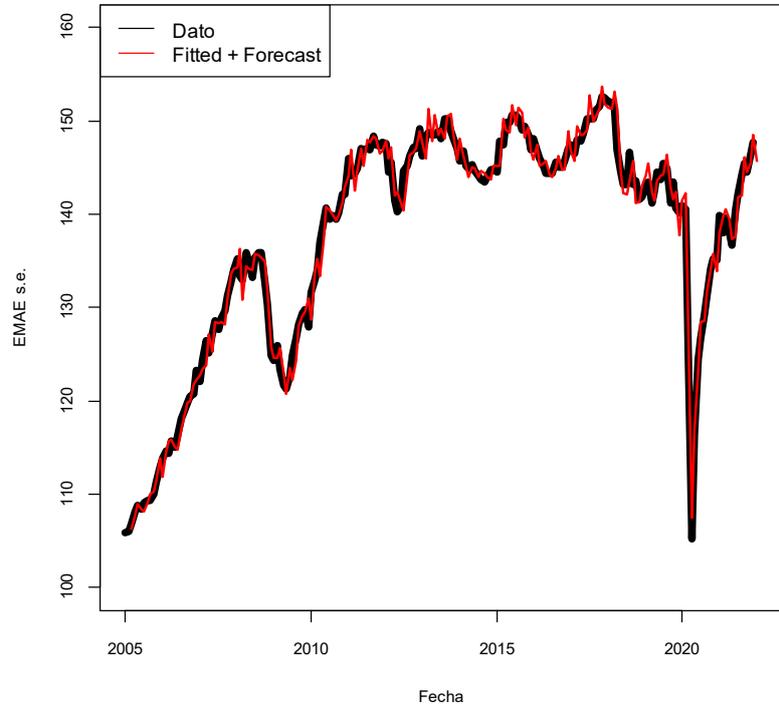
Una vez encontrado los hiper-parámetros óptimos de los modelos de *Machine Learning* y analizado el desempeño relativo de estos métodos con respecto a los modelos econométricos en el conjunto *Test*, se procede a proyectar el EMAE s.e. para enero de 2022 utilizando a toda la muestra como conjunto de entrenamiento y los parámetros (VAR, FAVAR) o hiper-parámetros (SSA, Jordan, LSTM) obtenidos de la sección 5.1. En este sentido, primero se presenta una serie de gráficos con los ajustes de los distintos modelos para todo el período de análisis y la proyección de enero de 2022. Luego, se presentan dos tablas: una con el RMSE y el MAPE de cada uno de los métodos de estimación entre 2005 y 2021 y otra con las proyecciones puntuales de enero de 2022.

Gráfico 20. Ajuste de los diferentes modelos para todo el período de análisis y proyección de enero de 2022

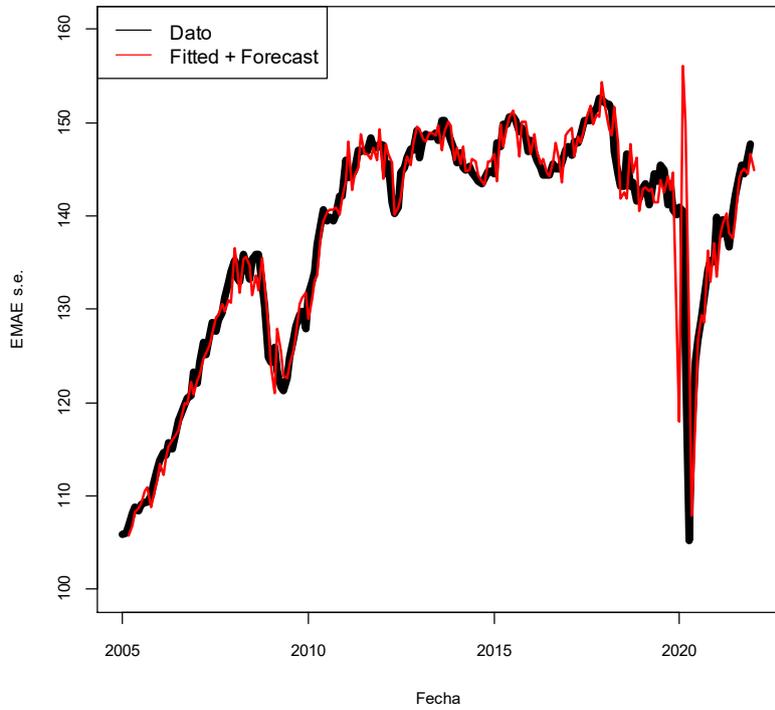
ARIMA - Ajuste del modelo para toda la muestra + Proyeccion enero 2022



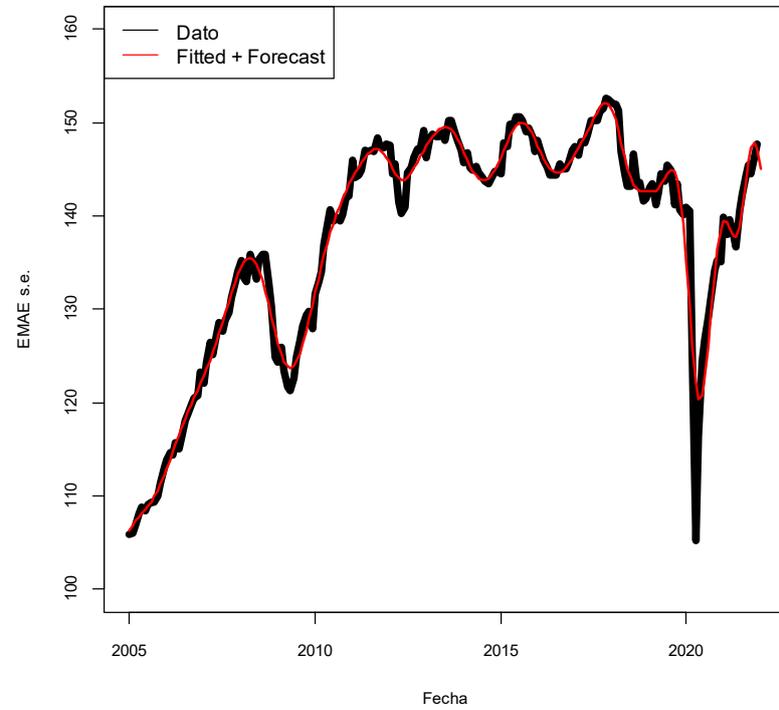
VAR - Ajuste del modelo para toda la muestra + Proyeccion enero 2022



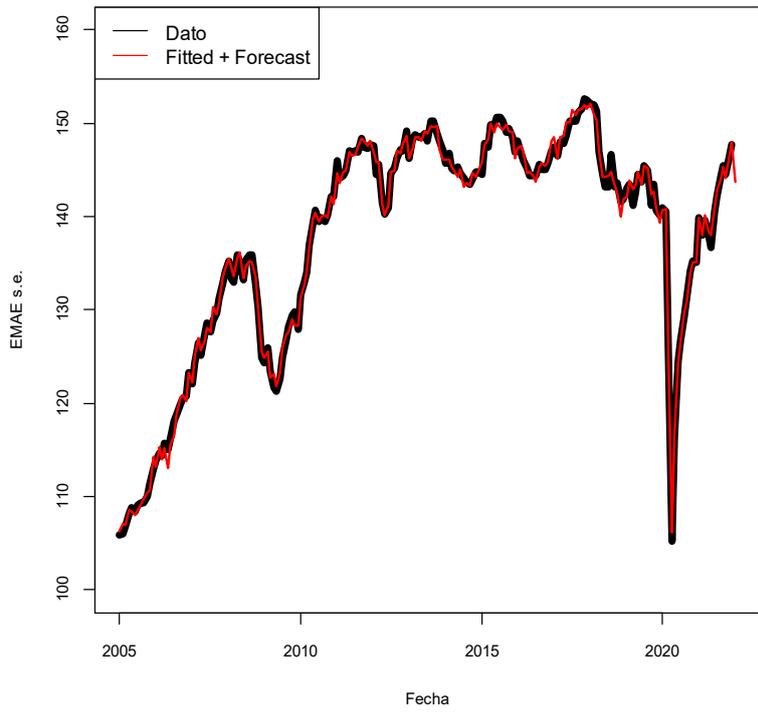
FAVAR - Ajuste del modelo para toda la muestra + Proyeccion enero 2022



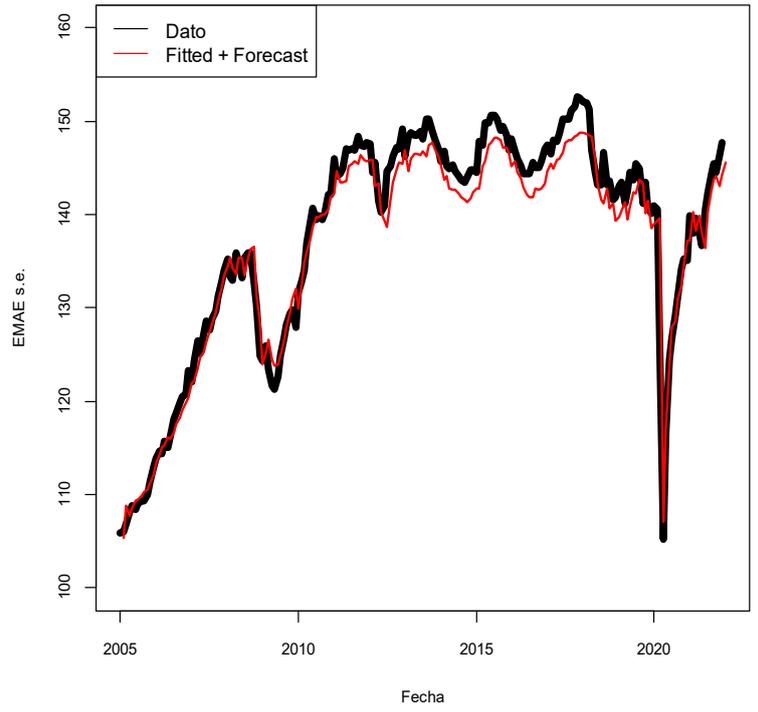
SSA - Ajuste del modelo en toda la muestra + Proyeccion enero 2022



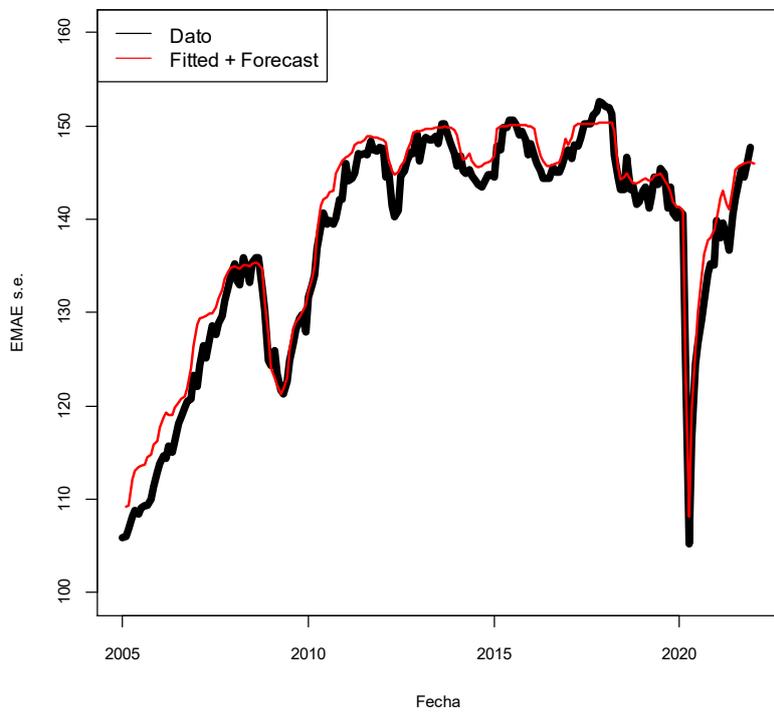
Jordan - Ajuste del modelo en toda la muestra + Proyeccion enero 2022



LSTM Univariado - Ajuste del modelo en toda la muestra + Proyeccion enero 2022



LSTM Multivariado - Ajuste del modelo en toda la muestra + Proyeccion enero 2022



Fuente: elaboración propia

Ante todo, vale la pena destacar que, en este caso, los resultados del modelo LSTM son exactamente iguales a los del LSTM *rolling*, tanto para el enfoque univariado

como para el enfoque multivariado. Esto se debe a que el conjunto *Train* es el mismo para ambos modelos dado que se utiliza toda la muestra para proyectar el EMAE s.e. en enero de 2022. Por este motivo, solo se presenta una especificación del LSTM univariado y LSTM multivariado.

En términos generales, todos los modelos presentan un buen ajuste a los datos, a excepción del modelo FAVAR que muestra un salto discreto en los meses correspondientes a la irrupción de la pandemia (aunque para el resto de los meses el ajuste es bueno). Los modelos ARIMA, VAR y Jordan muestran estimaciones muy buenas y similares en términos gráficos, donde las del último parecen ser levemente más suaves que las de los dos primeros. En relación a esto, el modelo SSA claramente suaviza los movimientos del EMAE s.e. y es menos sensible a *outliers*, por lo que apunta a ser un buen modelo para verificar la tendencia de la actividad económica. Por su parte, los modelos LSTM presentan un ajuste aceptable, aunque algo peor en relación a los otros modelos mencionados. En particular, entre 2012 y 2019 el enfoque univariado parece subestimar levemente el nivel de actividad, mientras que el enfoque multivariado lo sobreestima levemente.

Tabla 4. RMSE y MAPE de los distintos modelos para toda la muestra

	Train: 2005-2021			
	RMSE	Ranking	MAPE	Ranking
ARIMA	2.52	6	1.14	4
VAR	1.40	2	0.79	2
FAVAR	3.77	7	1.50	7
SSA	1.93	3	0.84	3
Jordan	0.67	1	0.39	1
LSTM Univariado	2.37		1.33	
LSTM Univariado <i>Rolling</i>	2.37	5	1.33	5
LSTM Multivariado	2.36		1.47	
LSTM Multivariado <i>Rolling</i>	2.36	4	1.47	6

Fuente: elaboración propia

Al comparar los resultados de la Tabla 3 y la Tabla 4, se puede verificar que el RMSE y el MAPE disminuye para todos los modelos producto de la mayor cantidad de

observaciones y a que las observaciones atípicas (marzo y abril de 2020) pesan menos sobre la cantidad de observaciones temporales totales en análisis (el 3,33% de las observaciones en la sección 5.1 y el 0,98% de las observaciones en la sección 5.2). Más allá de esto, también hay un cambio en el desempeño relativo de los modelos. En particular, al analizar el RMSE y el MAPE de forma conjunta, el modelo de Jordan se consolida como el más preciso, seguido por el VAR y el SSA. En particular, el modelo SSA es el que evidencia la mejora más fuerte entre el conjunto *Test* y toda la muestra, ya que escala de los últimos puestos a la tercera posición. A ellos le siguen los modelos ARIMA, LSTM Univariado y LSTM Multivariado con desempeños similares al analizar las dos funciones de pérdida, los cuales se reparten equitativamente el cuarto, quinto y sexto puesto en el *ranking*. En cambio, en el último lugar se encuentra el modelo FAVAR, producto de las erráticas predicciones en los meses en los que irrumpió la pandemia.

Tabla 5. Proyección de enero de 2022 de los distintos modelos y el EMAE s.e.

	ene-22	dic-21	% m/m s.e.
ARIMA	147.50	146.87	0.4%
VAR	145.65	148.52	-1.9%
FAVAR	144.93	146.58	-1.1%
SSA	145.08	147.13	-1.4%
Jordan	143.67	148.02	-2.9%
LSTM Univariado	145.57	144.20	0.9%
LSTM Univariado <i>Rolling</i>	145.57	144.20	0.9%
LSTM Multivariado	146.03	146.08	0.0%
LSTM Multivariado <i>Rolling</i>	146.03	146.08	0.0%
EMAE s.e.	146.93	148.03	-0.7%

Fuente: elaboración propia

Al analizar las proyecciones de enero de 2022, el [INDEC](#) publicó un EMAE s.e. de 146,93⁵ para el primer mes del año, cifra que marca un retroceso del 0,7% mensual desestacionalizado (m/m s.e.). Los modelos ARIMA y LSTM multivariado son aquellos que poseen valores más cercanos al reportado por el INDEC. A ellos le siguen el modelo

⁵ La serie del EMAE s.e. que publica el INDEC posee revisiones. Esta cifra se encuentra en el último informe publicado por el organismo, el cual corresponde al mes de marzo de 2022.

VAR, el LSTM Univariado y el modelo SSA. En cambio, los modelos FAVAR y de Jordan son los que poseen los valores más lejanos al reportado por el INDEC. En este aspecto, vale la pena realizar algunas aclaraciones. En primer lugar, en este punto solo se analizan las proyecciones de un mes determinado. Por lo tanto, los resultados son puntuales y no necesariamente indican que el modelo que produce el valor más cercano es el mejor ya que en otros meses los resultados podrían ser distintos. Para determinar el modelo más preciso, en las secciones 5.1 y 5.2 se analizan los errores de las estimaciones de los diferentes modelos para una mayor cantidad de observaciones temporales que un mes puntual. En particular, se analizó el RMSE y el MAPE de los diversos métodos de estimación para el conjunto *Test* (2017-2021) y para toda la muestra (2005-2021). En este punto, es importante recordar que el modelo de Jordan tuvo el mejor desempeño relativo al analizar toda la muestra (ver Tabla 4), mientras que en el conjunto *Test* (ver Tabla 3) fue el de mejor desempeño relativo según el RMSE y el segundo mejor según el MAPE.

Además, estos resultados probablemente estén influenciados por el hecho de que enero de 2022 fue un mes muy particular para la actividad económica. En mayor detalle, en el arranque de año hubo una serie de factores atípicos que causaron un retroceso de la economía argentina: el rebrote de contagios de COVID-19 a niveles récord, la menor cantidad de días hábiles en la industria producto de las paradas técnicas en las plantas (en la industria automotriz solo hubo 9 días hábiles cuando en enero de 2021 hubo 17), las altas temperaturas de la primera quincena que derivaron en una sequía y en numerosos cortes de luz y el impacto de las lluvias de la segunda quincena sobre la actividad de la construcción. Todos estos factores sugerían, a través de los indicadores sectoriales, que el retroceso de la actividad económica iba a ser más pronunciado en relación a lo que terminó reportando el INDEC.

En este marco, en la Tabla 5 se agrega el dato del EMAE s.e. correspondiente a diciembre de 2021 y las proyecciones de los diferentes modelos para dicho mes con el objetivo de verificar si los modelos registran correctamente la “dirección” en la que se movió la economía argentina en el arranque del 2022. Tal como se puede ver en la Tabla 5, ningún modelo con enfoque multivariado predice una expansión económica. En particular, los modelos VAR, FAVAR y Jordan proyectan caídas entre 1,1% y 2,9% m/m s.e., los cuales superan al dato reportado por el INDEC; mientras que el LSTM multivariado predice un estancamiento económico. En cambio, la mayoría de los modelos

univariados proyectan una expansión económica: +0,4% m/m s.e. para el modelo ARIMA y +0,9% m/m s.e. para el modelo LSTM univariado. Por su parte, vale la pena remarcar la estimación del modelo SSA, la cual marca una caída del 1,4% m/m s.e. desde un enfoque univariado. Este análisis cualitativo no es menor ya que refleja el valor agregado que brindan los enfoques multivariados con los indicadores económicos adelantados. En la mayoría de los casos probablemente sea más útil predecir en qué dirección se movió la economía antes que estimar el nivel de actividad en términos desestacionalizados.

5.3 Test de Diebold-Mariano

Si bien las funciones de pérdida brindaron diferentes resultados para los diversos métodos de estimación, también resulta interesante evaluar si las proyecciones de los diferentes modelos son estadísticamente diferentes entre sí. Para ello, se efectúa el *test* de Diebold-Mariano.

En términos teóricos, supongamos que tenemos dos proyecciones f_1, \dots, f_n y g_1, \dots, g_n para una serie temporal y_1, \dots, y_n y que $e_i = y_i - f_i$ y $r_i = y_i - g_i$ son los residuos de dichas proyecciones. La serie temporal $d_i = e_i^2 - r_i^2$ constituye la diferencia entre los residuos de las predicciones y el supuesto clave para el *test* de Diebold-Mariano es que esta serie es estacionaria. En este marco, para $n > k \geq 1$ definimos la autocovarianza del rezago k como $\gamma_k = \frac{1}{n} \sum_{i=k+1}^n (d_i - \bar{d})(d_{i-k} - \bar{d})$, donde $\bar{d} = \frac{1}{n} \sum_{i=1}^n d_i$.

Para $h \geq 1$, definimos al estadístico Diebold-Mariano (DM) como:

$$DM = \frac{\bar{d}}{\sqrt{[\gamma_0 + 2 \sum_{k=1}^{h-1} \gamma_k]/n}} \quad (32)$$

Bajo el supuesto de que $E[d_i] = 0$ (hipótesis nula del *test*), el estadístico DM sigue una distribución normal estándar: $DM \sim N(0,1)$. Por lo tanto, encontraremos una diferencia estadísticamente significativa entre las proyecciones si el valor absoluto del estadístico DM es mayor a cierto valor crítico que dependerá del valor de significación escogido, es decir, si rechazamos la hipótesis nula. En este sentido, en la Tabla 6 se

presentan los p-valores del test de Diebold-Mariano (el cual fue efectuado para todo el período de análisis) para todos los cruces posibles entre los diversos modelos utilizados en este trabajo. A un nivel de significación del 5%, un p-valor menor a 0,05 (celdas resaltadas en verde) indica que hay diferencias entre los residuos de las predicciones ya que se rechaza la hipótesis nula, lo que implica que las proyecciones son estadísticamente diferentes entre sí. En cambio, un p-valor mayor a 0,05 implica que las proyecciones no son estadísticamente diferentes entre sí. En este punto, aquellos valores entre 0,05 y 0,10 se encuentran resaltados en amarillo por su cercanía al nivel de significación escogido, mientras que aquellos valores superiores a 0,10 se encuentran resaltados en rojo.

Tabla 6. P-valores de los tests de Diebold-Mariano entre los diversos métodos de estimación

	ARIMA	VAR	FAVAR	SSA	Jordan	LSTM Uni	LSTM Multi
ARIMA							
VAR	0.03						
FAVAR	0.03	0.01					
SSA	0.06	0.23	0.01				
Jordan	0.00	0.00	0.00	0.03			
LSTM Uni	0.67	0.00	0.05	0.31	0.00		
LSTM Multi	0.70	0.00	0.08	0.25	0.00	0.95	

Fuente: elaboración propia

De la Tabla 6 se desprende que parece haber diferencias estadísticamente significativas en las proyecciones de la mayoría de los modelos. En particular, las proyecciones que genera el modelo de Jordan son estadísticamente diferentes a las que producen todo el resto de los métodos. El modelo FAVAR parece apuntar en la misma dirección, aunque con respecto a los dos modelos LSTM se verifica un p-valor levemente superior al 0,05. De forma contrapuesta, las predicciones del modelo LSTM univariado y multivariado son las más similares entre sí, mientras que también hay fuertes similitudes entre los pronósticos del modelo ARIMA y las estimaciones de los dos modelos LSTM. Por último, las proyecciones del modelo SSA poseen semejanzas estadísticas con las del modelo VAR y los dos modelos LSTM.

5.4 Extensión para modelos multivariados: estimaciones en “tiempo real”

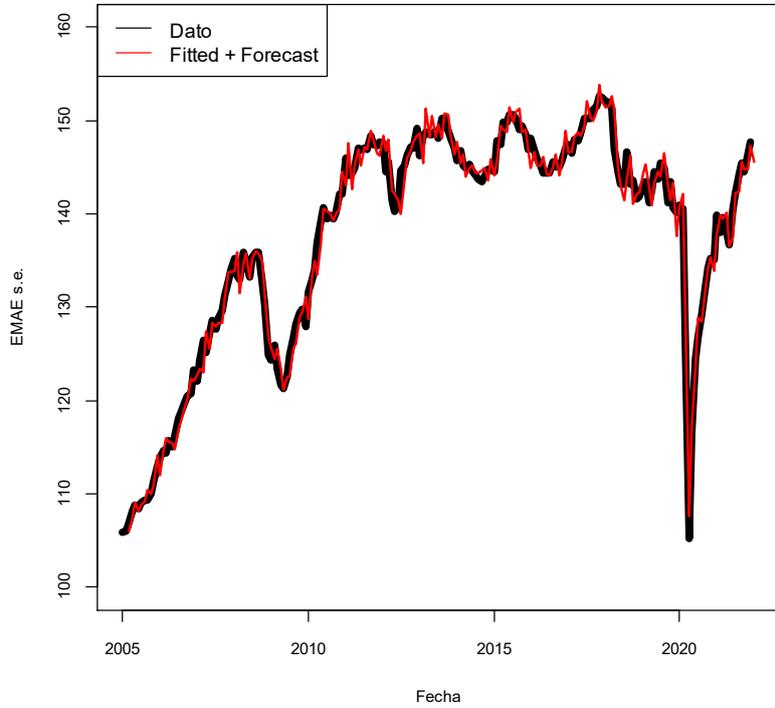
En relación a los modelos multivariados, las estimaciones incluyen indicadores que se conocen anticipadamente al EMAE, pero que también poseen un rezago considerable. Por este motivo, procedo a realizar nuevamente las estimaciones de los modelos multivariados (VAR⁶, FAVAR⁷, Jordan, LSTM multivariado) considerando únicamente a las variables que se conocen a los 20 días posteriores al cierre de cada mes. Entre ellas encontramos a 15 indicadores: patentamientos de automóviles, producción de automóviles, ventas de automóviles a concesionarios, exportaciones de automóviles, despachos de cemento, demanda de electricidad de grandes usuarios, índice Construya, recaudación tributaria ligada a la actividad, depósitos del sector privado en pesos, préstamos del sector privado en pesos, M2 privado, tipo de cambio real, índice Merval, confianza al consumidor y la variable binaria que contempla la irrupción de la pandemia. Vale la pena aclarar que en esta sección se replican los mismos modelos que en la sección 5.2 para que los resultados sean comparables, aunque, si se quisiera avanzar únicamente con estas versiones de los métodos de estimación, sería importante encontrar el rezago óptimo de los modelos VAR y FAVAR y los hiper-parámetros óptimos de los modelos de Jordan y LSTM multivariado. A continuación, se vuelve a presentar una serie de gráficos con los ajustes de los distintos modelos para todo el período de análisis y la proyección de enero de 2022, una tabla con el RMSE y el MAPE de cada uno de los modelos entre 2005 y 2021 y otra tabla con las proyecciones puntuales de enero de 2022.

⁶ En el modelo VAR se habían considerado algunas variables que se conocen después de los 20 días posteriores al cierre de mes: volumen de importaciones, la producción de petróleo y la producción de gas. En esta sección, el modelo VAR excluye dichas variables.

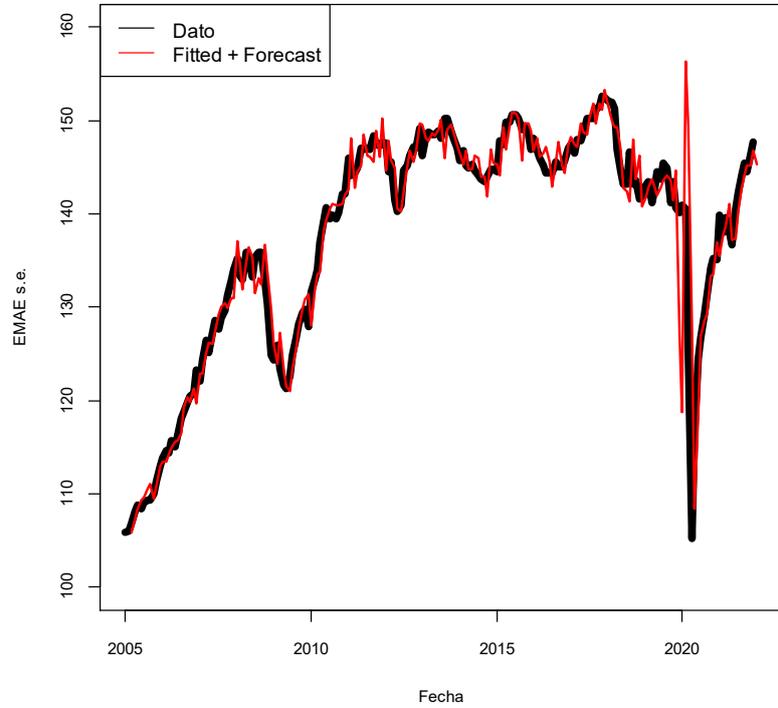
⁷ En el modelo FAVAR se vuelve a omitir la producción de automóviles porque la serie desestacionalizada produce un valor negativo.

Gráfico 21. Ajuste de los modelos multivariados en “tiempo real” para todo el período de análisis y proyección de enero de 2022.

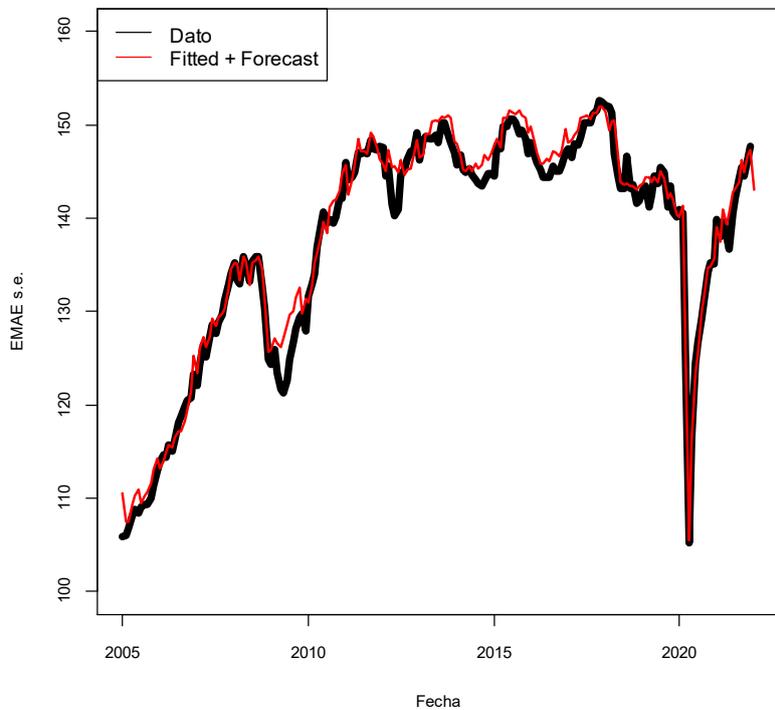
VAR - Ajuste del modelo para toda la muestra + Proyeccion enero 2022



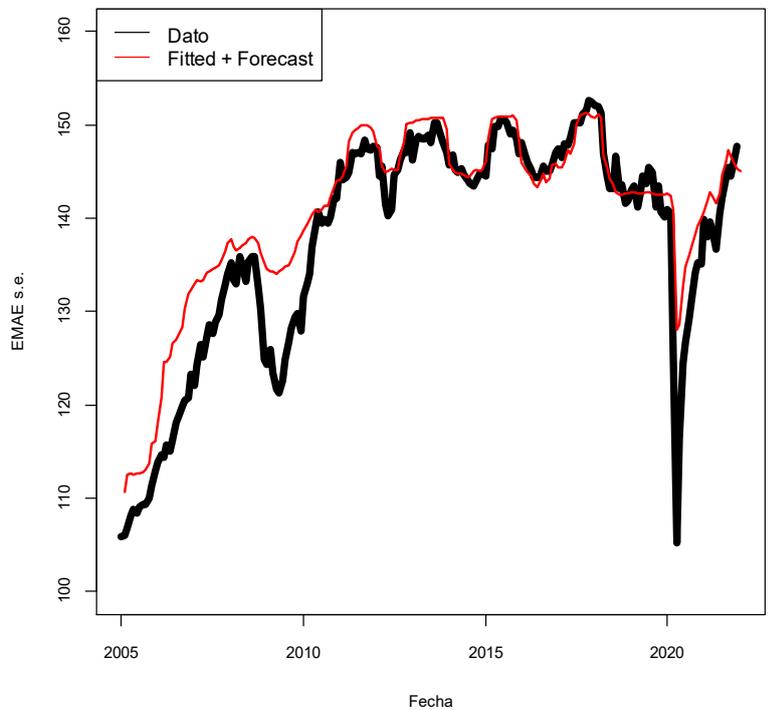
FAVAR - Ajuste del modelo para toda la muestra + Proyeccion enero 2022



Jordan - Ajuste del modelo en toda la muestra + Proyeccion enero 2022



LSTM Multivariado - Ajuste del modelo en toda la muestra + Proyeccion enero 2022



Fuente: elaboración propia

En términos gráficos, se puede observar que el ajuste de los modelos VAR y FAVAR son muy similares con respecto a la base de datos completa. En el caso del modelo de Jordan el ajuste se deteriora levemente, mientras que se verifica un desempeño claramente peor para el modelo LSTM multivariado ya que no logra captar con precisión la irrupción de la pandemia y sobreestima el nivel de actividad durante los primeros años de la muestra.

Tabla 7. RMSE y MAPE de los modelos multivariados en “tiempo real” (base de datos reducida) y con la base de datos completa para toda la muestra

	Base de datos reducida		Base de datos completa	
	RMSE	MAPE	RMSE	MAPE
VAR	1.49	0.84	1.40	0.79
FAVAR	3.71	1.46	3.77	1.50
Jordan	1.69	0.94	0.67	0.39
LSTM Multivariado	4.90	2.69	2.36	1.47

Fuente: elaboración propia

Al analizar el RMSE y el MAPE de los modelos con la base de datos reducida y completa, las intuiciones obtenidas desde los gráficos se mantienen. Por un lado, se puede observar que las funciones de pérdida son muy similares para los modelos econométricos tradicionales: empeora marginalmente en el caso del modelo VAR y mejora levemente en el caso del modelo FAVAR. Por otro lado, en el modelo de Jordan las funciones de pérdida son algo mayores con la base de datos reducida y en el modelo LSTM multivariado son claramente mayores. Además, cabe destacar que, con la base de datos reducida, el modelo de Jordan y el modelo VAR siguen siendo los más precisos, mientras que el FAVAR pasa a tener un mejor desempeño en relación al LSTM multivariado.

Tabla 8. Proyección de enero de 2022 de los modelos multivariados en “tiempo real”
(base de datos reducida) y con la base de datos completa y el EMAE s.e.

	Base de datos reducida			Base de datos completa		
	ene-22	dic-21	% m/m s.e.	ene-22	dic-21	% m/m s.e.
VAR	145.57	147.25	-1.1%	145.65	148.52	-1.9%
FAVAR	145.28	146.81	-1.0%	144.93	146.58	-1.1%
Jordan	142.99	147.24	-2.9%	143.67	148.02	-2.9%
LSTM Multivariado	145.07	145.34	-0.2%	146.03	146.08	0.0%
EMAE s.e.	146.93	148.03	-0.7%	146.93	148.03	-0.7%

Fuente: elaboración propia

Al analizar la proyección de enero de 2022 con la base de datos reducida, todos los modelos marcan un retroceso de la actividad económica, al igual que el EMAE s.e.; aunque siguen subestimando los niveles de actividad, al igual a lo que sucedía con la base de datos completa. En particular, las variaciones mensuales son similares a las proyectadas con la base de datos completa, incluso se acercan un poco más a lo reportado por el INDEC.

Por lo tanto, reducir la base de datos a aquellos indicadores que se conocen para los primeros días posteriores al cierre de cada mes puede ser una herramienta útil para proyectar la evolución de la actividad económica con un menor rezago, ya que la calidad del ajuste de los modelos VAR y de Jordan sigue siendo muy bueno para todo el período de análisis y, si no contemplamos los meses de irrupción de la pandemia, el modelo FAVAR también ajusta muy bien. Además, tal como se mencionó anteriormente, hay que tener en cuenta que, para estos modelos multivariados en “tiempo real”, aun restaría optimizar los rezagos de los modelos econométricos tradicionales y los hiper-parámetros de los modelos de *Machine Learning*, lo que podría derivar en resultados más precisos a los que se exponen en esta sección.

6. Conclusiones

En este trabajo se busca proyectar la evolución de la actividad económica argentina (medida a través del EMAE) a un mes vista con modelos de *Machine Learning*, tanto desde enfoques univariados como multivariados, cuyos resultados son contrastados con modelos econométricos más tradicionales de series de tiempo.

En términos muy generales, se observa un buen desempeño de los modelos, incluso en períodos de *shocks* muy fuertes como la irrupción de la pandemia a comienzos de 2020, lo que supone un gran desafío al momento de efectuar las estimaciones. En este punto, hay que tener en cuenta que en el bimestre marzo-abril de 2020 la economía argentina marcó un retroceso del 25% con respecto a febrero, fenómeno inédito para la historia de nuestro país. Al evaluar los resultados de los distintos modelos por separado, se puede concluir que los algoritmos de *Machine Learning* constituyen herramientas útiles para predecir la evolución de la economía argentina en el corto plazo. De hecho, los métodos de aprendizaje supervisado muestran, en términos generales, desempeños superiores a los modelos econométricos tradicionales ya que producen errores de menor magnitud. Este resultado se encuentra en línea con las conclusiones de algunos trabajos que buscan predecir, con métodos de aprendizaje supervisado, la evolución de la actividad económica de diferentes países.

Además, a la hora de proyectar el EMAE s.e. parece conveniente adoptar enfoques multivariados ya que estos contemplan una serie de indicadores económicos que contienen información parcial adelantada sobre la evolución de la actividad, donde algunos de estas variables también poseen una elevada correlación con el EMAE s.e. La principal ventaja de los modelos multivariados radica en que permiten estimar con mayor precisión la dirección en la que se movió la economía argentina en un mes determinado, tal como se mostró en el apartado 5.2 de este trabajo.

En este marco, se puede concluir que el modelo de Jordan parece brindar las proyecciones estadísticamente más precisas ya que (i) es el método que produce los menores errores de predicción medidos a través del RMSE y el MAPE y (ii) genera, según el *test* de Diebold-Mariano, pronósticos estadísticamente diferentes con respecto a las estimaciones de todo el resto de los modelos. Una vez aclarado esto, el modelo VAR

también produce buenos resultados para todo el período de análisis al ubicarse en el segundo puesto según las dos funciones de pérdida. Por su parte, el modelo FAVAR también ajusta de forma similar al VAR a lo largo de toda la muestra, pero no logra capturar de manera adecuada los grandes *shocks*, tal como sucede con la irrupción de la pandemia. Por lo tanto, el modelo FAVAR puede resultar un modelo útil si se considera que en el futuro cercano la economía argentina no va a experimentar un *shock* de elevada magnitud. Por último, si bien el modelo LSTM multivariado ajusta bien en todo el período de análisis, presenta estimaciones poco precisas en el conjunto *Test* y con la base de datos reducida, por lo que no resulta un método del todo fiable.

En cambio, dentro de los enfoques univariados, vale la pena destacar el desempeño del modelo SSA. A diferencia del resto de los métodos de estimación, el SSA suaviza la evolución de la actividad económica, es robusto a *outliers* y predice un retroceso de la economía cuando se evalúa la proyección de enero de 2022. Por su parte, el modelo LSTM Univariado muestra un buen desempeño ya que alcanza el podio cuando se evalúan las predicciones en el conjunto *Test* y brinda errores de menor magnitud que el modelo ARIMA; aunque en toda la muestra se ubica en la quinta posición y exhibe resultados muy similares a los del modelo ARIMA. Por lo tanto, se podría concluir que el modelo LSTM Univariado es, al menos, tan bueno como el modelo ARIMA.

En definitiva, para proyectar el EMAE s.e. a un mes vista con cierta precisión y celeridad se debería seleccionar el modelo de Jordan. Esto se debe a dos factores principales: (i) es el método de estimación más preciso y (ii) mantiene un buen desempeño cuando se estima en “tiempo real”, es decir, a los 20 días de finalizado el mes. A los fines de tener modelos *benchmark*, se debería escoger a los modelos VAR y SSA. Por un lado, la selección del modelo VAR se debe a que es el segundo mejor modelo cuando se analizan las funciones de pérdida para toda la muestra y a que mantiene un desempeño prácticamente idéntico al efectuar las estimaciones en “tiempo real”. Por otro lado, el modelo SSA es menos sensible a valores atípicos y es un modelo que captura muy bien la tendencia de la actividad económica.

Sin embargo, hay que tener en cuenta que el desempeño relativo de los diversos métodos de estimación puede ir cambiando a lo largo del tiempo, tal como se observó con los cambios en el *ranking* de los modelos al evaluar las funciones de pérdida en el

conjunto *Test* y en todo el período de análisis. A modo extensivo de este trabajo, puede resultar útil implementar los modelos de *Machine Learning* en otras series económicas o financieras de la economía argentina, tal como la inflación, el tipo de cambio financiero y/o paralelo y los precios o tasas de diversos activos financieros (bonos, créditos corporativos y acciones).

7. Referencias bibliográficas

- Balkin, S. D.; Keith, O. J. (2000). *Automatic neural network modeling for univariate time series*. International Journal of Forecasting. Vol. 16, No. 4, pp. 509-515.
Disponible en <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.137.8386&rep=rep1&type=pdf>.
- BCRA (2018). *Informe de Política Monetaria - Abril 2018*. Disponible en http://bcra.gov.ar/PoliticaMonetaria/IPOM0418.asp#apartado_2.
- BCRA (2020). *Nowcast de Actividad en el Banco Central*. Disponible en <https://centraldeideas.blog/nowcast-de-actividad-en-el-banco-central/#:~:text=El%20fundamento%20b%C3%A1sico%20del%20Nowcasting,diarias%2C%20semanales%2C%20mensuales>.
- Blanco, E.; D'Amato, L.; Dogliolo, F.; Garegnani, L. (2017). *Nowcasting GDP in Argentina: Comparing the predictive ability of different models*, BCRA Investigaciones Económicas. Economic Research Working Paper N° 74. Disponible en <http://www.bcra.gov.ar/Institucional/DescargaPDF/DownloadPDF.aspx?Id=594>
- Briceño, H.; Rocco, C.; Zio, E. (2013). *Singular Spectrum Analysis for Forecasting of Electric Load*. AIDIC. Vol. 33. Disponible en <https://www.aidic.it/cet/13/33/154.pdf>
- Chuku, C.; Oduor, J.; Simpasa, A. (2019). *Intelligent forecasting of economic growth for African economies: Artificial neural networks versus time series and structural econometric models*. International Economics. Vol. 159, No. 4. Disponible en <https://unassumingeconomist.com/wp-content/uploads/2017/04/Intelligent-forecast-using-ANN.pdf>
- D'Amato, L.; Garegnani, L.; Blanco, E. (2016). *Nowcasting de PBI: evaluando las condiciones cíclicas de la economía argentina*. BCRA Ensayos Económicos. No. 74, pp. 7-26. Disponible en <http://www.bcra.gov.ar/Institucional/DescargaPDF/DownloadPDF.aspx?Id=322>
- Drchal, J. (2006). *Evolution of Recurrent Neural Networks*. Czech Technical University in Prague. Disponible en https://www.researchgate.net/publication/223999346_Evolution_of_Recurrent_Neural_Networks
- Du, K.; Swamy, M. (2013). *Neural Networks and Statistical Learning*. Concordia University, Ch. 11, pp 351-363. Disponible en https://www.researchgate.net/publication/264912450_Neural_Networks_and_Statistical_Learning
- Golyandina, N. (2010). *On the choice of parameters in singular spectrum analysis and related subspace-based methods*. Statistics and Its Interface. Vol. 3, pp 259-279. Disponible en <https://arxiv.org/pdf/1005.4374.pdf>

Golyandina, N. (2021). *Particularities and commonalities of singular spectrum analysis as a method of time series analysis and signal processing*. WIREs Computational Statistics. Vol. 12, No. 4. Disponible en <https://arxiv.org/pdf/1907.02579.pdf>

Golyandina, N.; Korobeynikov, A. (2014). *Basic Singular Spectrum Analysis and Forecasting with R*. Computational Statistics and Data Analysis. Vol 71, pp 934-954. Disponible en <https://arxiv.org/pdf/1206.6910.pdf>

Golyandina, N., & Zhigljavsky, A. (2013). *Singular Spectrum Analysis for Time Series*. Ch. 2. Disponible en https://www.researchgate.net/publication/260124592_Singular_Spectrum_Analysis_for_Time_Series

Hassani, H. (2007). *Singular Spectrum Analysis: Methodology and Comparison*. Journal of Data Science. Vol. 5, No. 4991, pp. 239-257. Disponible en <https://www.jds-online.com/files/JDS-396.pdf>

Hassani, H. (2009). *A Brief Introduction to Singular Spectrum Analysis*. Cardiff University. Disponible en http://ssa.cf.ac.uk/ssa2010/a_brief_introduction_to_ssa.pdf

Hassani, H.; Rua, A.; Silva, E.; Thomakos, D. (2019). *Monthly Forecasting of GDP with Mixed Frequency Multivariate Singular Spectrum Analysis*. International Journal of Forecasting. Vol 35, No. 4, pp. 1263-1272. Disponible en https://ualresearchonline.arts.ac.uk/id/eprint/15511/1/2019__1_.pdf

Heravi, S.; Osborn, D. R.; Birchenhall, C. R. (2004). *Linear versus neural network forecasts for European industrial production series*. International Journal of Forecasting. Vol. 20, No. 3, pp. 435-446. Disponible en https://www.researchgate.net/publication/222581913_Linear_versus_Neural_Network_Forecasts_for_European_Industrial_Production_Series_by

IBM (2021). *El modelo de redes neuronales*. Disponible en <https://www.ibm.com/docs/es/spss-modeler/SaaS?topic=networks-neural-model>

INDEC (2016). *Cuentas Nacionales Metodología de estimación*. Disponible en https://www.indec.gov.ar/ftp/cuadros/economia/metodologia_21_cuentas_nacionales.pdf

INDEC (2016). *Metodología del Estimador Mensual de Actividad Económica (EMAE)*. Disponible en https://www.indec.gov.ar/ftp/cuadros/economia/metodologia_ema_e_ago_16.pdf

INDEC (2020). *Preguntas frecuentes sobre el Sistema de Cuentas Nacionales*. Disponible en https://www.indec.gov.ar/ftp/cuadros/economia/preguntas_frecuentes_cuentas_nacionales_nota_5.pdf

Instituto Nacional de Estadística de España (2019). *Manual de usuario de JDemetra+*. Disponible en https://www.ine.es/clasifi/manual_jdemetra.pdf

- Interactive Chaos (s.f.). *Redes Neuronales*. Disponible en <https://interactivechaos.com/es/manual/tutorial-de-machine-learning/redes-neuronales>
- Jahn, M. (2018). *Artificial neural network regression models: Predicting GDP growth*. Hamburg Institute of International Economics. No. 185. Disponible en <https://www.econstor.eu/bitstream/10419/182108/1/1030440859.pdf>
- Jansen, A. (2021). *Sales forecasting of luxury products using LSTM neural networks and ARIMA models*. University of Groningen.
- Lien Oskarsson, M.; & Lin, C. (2018). *A simplified approach in FAVAR estimation*. Uppsala Universitet. Disponible en <https://www.diva-portal.org/smash/get/diva2:1215768/FULLTEXT01.pdf>
- Loermann, J.; Maas, B. (2019). *Nowcasting US GDP with artificial neural networks*. Munich Personal RePEc Archive from University Library of Munich. No. 94459. Disponible en https://mpra.ub.uni-muenchen.de/95459/1/MPRA_paper_95459.pdf
- Longo, L.; Riccaboni, M.; Rungi, A. (2022). *A neural network ensemble approach for GDP forecasting*. Journal of Economic Dynamics and Control. Vol. 134. Disponible en <https://www.sciencedirect.com/science/article/abs/pii/S016518892100213X>
- Olah, C. (2015). *Understanding LSTM Networks*. Disponible en <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- Olaniyi, S. M.; Ibina, E.; Eze, E. O. (2014). *Neural network applications in the forecasting of GDP of Nigeria as a function of key stock market indicators*. Pelagia Research. Vol. 5, No. 3, pp 204-212. Disponible en <https://www.primescholars.com/articles/neural-network-applications-in-the-forecasting-of-gdp-of-nigeria-as-a-function-of-key-stock-market-indicators.pdf>
- Onnen, H. (2021). *Temporal Loops: Intro to Recurrent Neural Networks for Time Series Forecasting in Python*. Disponible en <https://towardsdatascience.com/temporal-loops-intro-to-recurrent-neural-networks-for-time-series-forecasting-in-python-b0398963dc1f>
- Paccagnini, A. (2017). *Forecasting with FAVAR: macroeconomic versus financial factors*. Narodowy Bank Polski. No. 256. Disponible en https://www.nbp.pl/publikacje/materialy_i_studia/256_en.pdf
- Petneházi, G. (2019). *Recurrent Neural Networks for Time Series Forecasting*. Univeristy of Debrecen. Disponible en <https://arxiv.org/pdf/1901.00069.pdf>
- Phi, M. (2018). *Illustrated Guide to LSTM's and GRU's: A step by step explanation*. Disponible en <https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21>
- Phi, M. (2018). *Illustrated Guide to Recurrent Neural Networks*. Disponible en <https://towardsdatascience.com/illustrated-guide-to-recurrent-neural-networks-79e5eb8049c9>

Richardson, A.; Van Florenstein Mulder, T.; Vehbi, T. (2018). *Nowcasting New Zealand GDP using machine learning algorithms*. International Journal of Forecasting. Vol. 37, No. 2, pp. 941-948. Disponible en <https://www.sciencedirect.com/science/article/abs/pii/S016920702030159X#:~:text=Our%20results%20show%20that%20machine,Reserve%20Bank%20of%20New%20Zealand.>

Sanusi, N. A.; Moosin, A. F.; Kusairi, S. (2020). *Neural Network Analysis in Forecasting the Malaysian GDP*. Journal of Asian Finance, Economics and Business. Vol. 7, No. 12, pp. 109-114. Disponible en <https://www.koreascience.or.kr/article/JAKO202034651879122.pdf>

Sepúlveda Pinto, P. I. (2011). *Redes Neuronales Recurrentes con algoritmo de optimización por enjambre de partículas para pronóstico multivariado de captura de anchovetas en el norte de Chile*. Pontificia Universidad Católica de Valparaíso. Disponible en http://opac.pucv.cl/pucv_txt/txt-3000/UCF3405_01.pdf

Sunitha, G.; Sampath Kumar, K.; Jyothirani, S.; Haragopal, V. (2018). *Forecasting GDP using ARIMA and Artificial Neural Networks Models under Indian Environment*. International Journal of Mathematics Trends and Technology. Vol. 56, No. 1, pp. 60-70. Disponible en https://www.researchgate.net/publication/325980510_Forecasting_GDP_using_ARIMA_and_Artificial_Neural_Networks_Models_under_Indian_Environment

Tang, Z.; De Almeida, C.; Fishwick, P. (1991). *Time series forecasting using neural networks vs box-jenkins methodology*. Simulation, Vol. 57, No. 5, pp. 303-310. Disponible en <https://journals.sagepub.com/doi/abs/10.1177/003754979105700508?journalCode=simb>

Tkacz, G.; Hu, S. (1999). *Forecasting GDP growth using artificial neural networks*. Bank of Canada. Working paper 99-3. Disponible en <https://www.bankofcanada.ca/wp-content/uploads/2010/05/wp99-3.pdf>

Zhang, G. P.; Qi, M. (2005). *Neural network forecasting for seasonal and trend time series*. European Journal of Operational Research. Vol. 160, No. 2, pp. 501-514. Disponible en https://www.researchgate.net/publication/4871997_Neural_network_forecasting_for_seasonal_and_trend_time_series

8. Anexo

Tabla 1. RMSE de los hiper-parámetros del modelo SSA

L	M	RMSE	L	M	RMSE	L	M	RMSE
50	2	10.26	50	10	7.32	50	18	8.63
70	2	9.13	70	10	6.21	70	18	9.47
90	2	8.65	90	10	8.01	90	18	8.76
110	2	8.70	110	10	6.26	110	18	6.70
130	2	9.39	130	10	5.60	130	18	6.13
50	4	8.43	50	12	7.90	50	20	9.54
70	4	8.78	70	12	7.00	70	20	9.89
90	4	8.58	90	12	7.60	90	20	8.75
110	4	8.59	110	12	5.55	110	20	6.68
130	4	8.02	130	12	6.32	130	20	6.34
50	6	7.20	50	14	8.76			
70	6	8.40	70	14	8.70			
90	6	7.87	90	14	7.53			
110	6	7.08	110	14	6.37			
130	6	7.29	130	14	6.66			
50	8	6.19	50	16	8.86			
70	8	7.92	70	16	8.81			
90	8	8.24	90	16	8.28			
110	8	6.91	110	16	6.78			
130	8	6.27	130	16	6.63			

Tabla 1bis. RMSE de los hiper-parámetros del modelo SSA

L	M	RMSE	L	M	RMSE
95	10	7.70	95	12	7.72
100	10	7.45	100	12	7.18
105	10	6.51	105	12	6.11
110	10	6.26	110	12	5.55
115	10	6.16	115	12	5.58
120	10	6.03	120	12	5.76
125	10	5.73	125	12	6.45
95	11	8.31			
100	11	7.14			
105	11	6.23			
110	11	5.86			
115	11	5.82			
120	11	5.53			
125	11	5.75			

Tabla 2. RMSE de los hiper-parámetros del modelo Jordan

k	λ	RMSE
1	0.1	6.83
2	0.1	2.71
3	0.1	2.47
4	0.1	3.28
5	0.1	3.81
7	0.1	3.70
1	0.5	6.15
2	0.5	4.12
3	0.5	5.59
4	0.5	4.94
5	0.5	5.60
7	0.5	4.70

Tabla 3. RMSE de los hiper-parámetros del modelo LSTM Univariado

neuronas	epochs	drop	RMSE	neuronas	epochs	drop	RMSE	neuronas	epochs	drop	RMSE
10	10	0.5	8.75	10	10	0.1	7.64	10	10	0.01	7.51
20	10	0.5	7.65	20	10	0.1	6.88	20	10	0.01	5.67
50	10	0.5	9.04	50	10	0.1	9.92	50	10	0.01	9.99
100	10	0.5	9.92	100	10	0.1	9.86	100	10	0.01	9.90
200	10	0.5	9.93	200	10	0.1	9.95	200	10	0.01	9.77
300	10	0.5	9.90	300	10	0.1	9.54	300	10	0.01	9.17
10	20	0.5	6.61	10	20	0.1	5.64	10	20	0.01	5.35
20	20	0.5	6.23	20	20	0.1	6.03	20	20	0.01	6.16
50	20	0.5	7.90	50	20	0.1	10.09	50	20	0.01	7.23
100	20	0.5	10.12	100	20	0.1	10.05	100	20	0.01	10.08
200	20	0.5	10.07	200	20	0.1	8.76	200	20	0.01	9.65
300	20	0.5	10.03	300	20	0.1	9.94	300	20	0.01	9.28
10	50	0.5	5.28	10	50	0.1	4.41	10	50	0.01	3.95
20	50	0.5	5.93	20	50	0.1	6.93	20	50	0.01	6.49
50	50	0.5	9.04	50	50	0.1	5.37	50	50	0.01	10.46
100	50	0.5	10.16	100	50	0.1	5.35	100	50	0.01	9.96
200	50	0.5	10.16	200	50	0.1	9.78	200	50	0.01	9.69
300	50	0.5	10.16	300	50	0.1	10.01	300	50	0.01	9.27
10	100	0.5	5.16	10	100	0.1	4.13	10	100	0.01	3.65
20	100	0.5	7.88	20	100	0.1	6.57	20	100	0.01	7.85
50	100	0.5	9.83	50	100	0.1	4.19	50	100	0.01	7.26
100	100	0.5	10.15	100	100	0.1	6.03	100	100	0.01	5.87
200	100	0.5	9.82	200	100	0.1	10.16	200	100	0.01	9.83
300	100	0.5	9.60	300	100	0.1	9.95	300	100	0.01	8.76
10	150	0.5	5.13	10	150	0.1	3.94	10	150	0.01	3.87
20	150	0.5	9.21	20	150	0.1	6.61	20	150	0.01	7.87
50	150	0.5	12.62	50	150	0.1	4.35	50	150	0.01	9.46
100	150	0.5	6.15	100	150	0.1	6.81	100	150	0.01	4.20
200	150	0.5	10.17	200	150	0.1	10.16	200	150	0.01	9.83
300	150	0.5	10.08	300	150	0.1	10.00	300	150	0.01	8.75
10	200	0.5	5.66	10	200	0.1	4.55	10	200	0.01	3.97
20	200	0.5	10.74	20	200	0.1	6.12	20	200	0.01	7.47
50	200	0.5	12.34	50	200	0.1	4.02	50	200	0.01	10.94
100	200	0.5	5.93	100	200	0.1	4.53	100	200	0.01	4.11
200	200	0.5	10.16	200	200	0.1	10.16	200	200	0.01	9.82
300	200	0.5	10.13	300	200	0.1	9.92	300	200	0.01	8.75
10	250	0.5	5.79	10	250	0.1	4.33	10	250	0.01	4.17
20	250	0.5	11.99	20	250	0.1	5.14	20	250	0.01	7.26
50	250	0.5	13.58	50	250	0.1	4.43	50	250	0.01	13.78
100	250	0.5	5.53	100	250	0.1	4.14	100	250	0.01	4.04
200	250	0.5	10.16	200	250	0.1	10.16	200	250	0.01	9.82
300	250	0.5	9.88	300	250	0.1	9.97	300	250	0.01	8.76

Tabla 4. RMSE de los hiper-parámetros del modelo LSTM Multivariado

neuronas	epochs	drop	RMSE	neuronas	epochs	drop	RMSE	neuronas	epochs	drop	RMSE
10	10	0.5	9.72	10	10	0.1	10.80	10	10	0.01	13.47
20	10	0.5	11.40	20	10	0.1	10.65	20	10	0.01	10.06
50	10	0.5	17.59	50	10	0.1	16.51	50	10	0.01	13.48
100	10	0.5	13.41	100	10	0.1	11.36	100	10	0.01	15.09
200	10	0.5	10.69	200	10	0.1	6.00	200	10	0.01	8.79
300	10	0.5	10.14	300	10	0.1	9.60	300	10	0.01	8.77
10	20	0.5	10.58	10	20	0.1	16.31	10	20	0.01	17.97
20	20	0.5	16.02	20	20	0.1	17.39	20	20	0.01	12.82
50	20	0.5	18.61	50	20	0.1	21.64	50	20	0.01	14.87
100	20	0.5	17.57	100	20	0.1	10.77	100	20	0.01	7.32
200	20	0.5	12.99	200	20	0.1	4.84	200	20	0.01	6.69
300	20	0.5	9.37	300	20	0.1	9.47	300	20	0.01	8.93
10	50	0.5	12.91	10	50	0.1	18.14	10	50	0.01	17.85
20	50	0.5	19.29	20	50	0.1	19.95	20	50	0.01	11.48
50	50	0.5	20.50	50	50	0.1	15.73	50	50	0.01	12.45
100	50	0.5	25.60	100	50	0.1	13.66	100	50	0.01	6.94
200	50	0.5	9.07	200	50	0.1	4.52	200	50	0.01	15.42
300	50	0.5	8.80	300	50	0.1	8.88	300	50	0.01	9.37
10	100	0.5	16.23	10	100	0.1	14.69	10	100	0.01	16.36
20	100	0.5	20.65	20	100	0.1	20.06	20	100	0.01	12.17
50	100	0.5	16.79	50	100	0.1	14.79	50	100	0.01	11.53
100	100	0.5	16.39	100	100	0.1	13.19	100	100	0.01	8.13
200	100	0.5	8.13	200	100	0.1	4.43	200	100	0.01	15.04
300	100	0.5	4.49	300	100	0.1	8.86	300	100	0.01	8.87
10	150	0.5	16.26	10	150	0.1	7.64	10	150	0.01	15.21
20	150	0.5	20.32	20	150	0.1	15.81	20	150	0.01	13.65
50	150	0.5	20.63	50	150	0.1	18.35	50	150	0.01	11.58
100	150	0.5	15.29	100	150	0.1	9.14	100	150	0.01	7.73
200	150	0.5	11.08	200	150	0.1	6.02	200	150	0.01	11.51
300	150	0.5	5.48	300	150	0.1	8.96	300	150	0.01	11.35
10	200	0.5	18.25	10	200	0.1	14.10	10	200	0.01	17.09
20	200	0.5	17.42	20	200	0.1	13.98	20	200	0.01	16.89
50	200	0.5	17.04	50	200	0.1	15.81	50	200	0.01	10.35
100	200	0.5	18.25	100	200	0.1	11.97	100	200	0.01	13.06
200	200	0.5	11.62	200	200	0.1	6.25	200	200	0.01	5.70
300	200	0.5	9.51	300	200	0.1	9.10	300	200	0.01	13.68
10	250	0.5	15.33	10	250	0.1	10.84	10	250	0.01	16.98
20	250	0.5	17.33	20	250	0.1	15.27	20	250	0.01	15.03
50	250	0.5	22.47	50	250	0.1	16.07	50	250	0.01	11.45
100	250	0.5	17.78	100	250	0.1	10.35	100	250	0.01	9.65
200	250	0.5	9.35	200	250	0.1	5.92	200	250	0.01	7.18
300	250	0.5	5.94	300	250	0.1	9.08	300	250	0.01	8.47