RISK MANAGEMENT MAGAZINE

Vol. 16, Issue 3
September – December 2021

EXCERPT

https://www.aifirm.it/newsletter/progetto-editoriale/

Deep Learning for seasonality modelling in Inflation-Indexed Swap pricing

Pier Giuseppe Giribone, Duccio Martelli

Deep Learning for seasonality modelling in Inflation-Indexed Swap pricing

Pier Giuseppe Giribone (University of Genoa, Banca CARIGE) – Duccio Martelli (University of Perugia)

Article submitted to double-blind peer review, received on 3rd September 2021 and accepted on 24th November 2021

Abstract

An Inflation-Indexed Swap (IIS) is a derivative in which, at every payment date, the counterparties swap an inflation rate with a fixed rate. For the calculation of the Inflation Leg cash flows it is necessary to build a mathematical model suitable for the Consumer Price Index (CPI) projection. For this purpose, quants typically start by using market quotes for the Zero-Coupon swaps in order to derive the future trend of the inflation index, together with a seasonality model for capturing the typical periodical effects. In this study, we propose a forecasting model for inflation seasonality based on a Long Short Term Memory (LSTM) network: a deep learning methodology particularly useful for forecasting purposes. The CPI predictions are conducted using a FinTech paradigm, but in respect of the traditional quantitative finance theory developed in this research field. The paper is structured according to the following sections: the first two parts illustrate the pricing methodologies for the most popular IIS: the Zero Coupon Inflation-Indexed Swap (ZCIIS) and the Year-on-Year Inflation-Indexed Swap (YYIIS); section 3 deals with the traditional standard method for the forecast of CPI values (trend + seasonality), while section 4 describes the LSTM architecture, and section 5 focuses on CPI projections, also called inflation bootstrap. Then section 6 describes a robust check, implementing a traditional SARIMA model in order to improve the interpretation of the LSTM outputs; finally, section 7 concludes with a real market case, where the two methodologies are used for computing the fair-value for a YYIIS and the model risk is quantified.

Key Words:
Inflation-Indexed Swap (IIS), Year-on-Year Inflation Indexed Swap (YYIIS), Zero-Coupon Inflation-Indexed Swap (ZCIIS), Seasonality model, CPI bootstrap, Machine Learning (ML), Deep Learning, Long Short-Term Memory (LSTM) Network

1) Introduction

Inflation has been rising since the end of 2020, mainly due to the reopening of the economy after the pandemic crisis. Despite central banks suggesting that this type of inflation will be transitory, market participants seem to believe otherwise. Financial institutions and investors base their expectations on two main clues: on one hand, the several constraints that the international supply chain have to currently face, and on the other hand, the financial stimuli that governments from all over the world have adopted to encourage their national economies.

While nobody knows with certainty how high inflation will rise or how long it will persist, the expectations, that this round of inflation will not be so short as promised, have given investors an incentive for reconsidering financial products, able to protect the holder from the detrimental effects of the inflation; products that only in the recent past were almost disappeared.

The present paper is part of this strand of literature and it aims to shed light on the seasonality modeling in Inflation Indexed Swaps (IIS), a derivative contract in which, at every payment date, the counterparties swap an inflation rate with a fixed interest rate.

This study aims to extend the existing literature concerning the integration of machine learning in the field of quantitative finance. Machine learning methodologies are in fact increasingly spreading in the financial sector. Among the numerous examples of applications proposed by literature, the most popular ones are mainly aimed at solving the following problems:

- Input data quality (Pendyala, 2018)
- Innovative algo-trading techniques (De Prado, 2018)
- Optimal portfolio management (Heaton *et al.*, 2017)
- Pattern recognition and classification (Kim, 2017)
- Financial time-series forecasting as an alternative to traditional econometric approaches (ARIMA, Bayesian VAR, GARCH) (Mammadi, 2017; Yanui, 2017)

It is more difficult to find evidence in the literature of artificial intelligence methodologies applied to exotic financial instruments pricing or about the integration of traditional quantitative finance theory with the new FinTech methodologies. The traditional implementation regards the numerical solution of the so-called fundamental Black-Scholes-Merton PDE through Radial Basis Functions (Company *et al.*, 2018). Only more recently, the application of Regressive Neural Networks together with Monte Carlo method was suggested for evaluating early-exercise features in American and Bermuda option pricing in accordance with Longstaff-Schwartz methodology (Lelong and Lapeyre, 2020).

While the number of studies adopting machine learning techniques in the field of finance has grown quite rapidly, the field of research focusing on the identification of the seasonality effects with such methodologies is still limited. The research aims to bridge this gap proposing an innovative approach, through the design of a so-called long short-term memory (LSTM) network for the identification of the seasonality effects. In particular, the study implements a deep learning methodology able to consider potential highly nonlinear relationships between the values of the Consumer Price Index (CPI), sampled in the previous five years in accordance with the most common market standard convention. In fact. Bloomberg® and other well-known info-providers set this parameter equal to five years in their valuation platforms (e.g., the Bloomberg® SWIL and SWPM pricing modules). This setting is also adopted in the design of the LSTM network with the aim of maintaining this trading practice and of helping the interpretation of the results using the same market conventions. It is worth to note that the frequency of the sample in the considered time series is monthly: this is also a standard choice for the analysis of this index, given that it reflects the publishing time interval.

2) The pricing framework

An Inflation-Indexed Swap (IIS) is a swap deal in which, for each payment date, $T_1, \dots, T_M$, counterparty A pays to counterparty B the inflation rate in the considered period, while counterparty B pays to counterparty A the fixed rate. The inflation rate is calculated as the percentage return of the CPI over the reference time interval.

There are two main types of IIS traded on the market: the Zero-Coupon Inflation-Indexed Swap (ZCIIS) and the Year-on-Year Inflation-Indexed Swap (YYIIS) (Brigo and Mercurio, 2006).

In a ZCIIS, at the maturity date T_M, assuming $T_M = M$ years, counterparty B pays to counterparty A the fixed quantity:

$$\boldsymbol{N[(1+K)^M - 1]} \quad (1)$$

Where K and N are the fixed interest rate and the principal, respectively.
In return for this fixed payment, at the maturity date T_M, counterparty A pays to counterparty B the floating amount:

$$\boldsymbol{N\left[\frac{I(T_M)}{I_0} - 1\right]} \quad (2)$$

Where $\boldsymbol{I_0}$ is the reference CPI and $\boldsymbol{I(T_M)}$ is the value of the index at time $\boldsymbol{T_M}$.
In a YYIIS, for each payment date T_i, counterparty B pays to counterparty A the fixed amount:

$$\boldsymbol{N\varphi_i K} \quad (3)$$

Where φ_i is the year fraction of the fixed swap leg in the range $[T_{i-1}, T_i]$, $T_0 \coloneqq 0$ and N is the principal of the deal.
Counterparty A pays to counterparty B the floating amount equal to:

$$\boldsymbol{N\varphi_i\left[\frac{I(T_i)}{I(T_{i-1})} - 1\right]} \quad (4)$$

ZCIIS and YYIIS are typically quoted in terms of the corresponding equivalent fixed rate K.

2.1) Zero-Coupon Inflation-Indexed Swap (ZCIIS) pricing

The standard arbitrage-free pricing theory leads to the estimation of the fair value for a ZCIIS inflation leg at time t, $0 \le t \le T_M$ (Brigo and Mercurio, 2006):

$$\boldsymbol{ZCIIS(t, T_M, I_0, N) = N \cdot E_n\left\{exp\left(-\int_t^{T_M} n(u)du\right)\left[\frac{I(T_M)}{I_0} - 1\right]\middle|\, \mathrm{F}_t\right\}} \quad (5)$$

Where F_t is the σ-algebra generated by the stochastic process of the underlying up to time t.
The nominal price of a zero coupon bond is equal to the price of a contract that pays one unit of the CPI Index at bond maturity. In formulas, for each $t < T$:

$$\boldsymbol{I(t)P_r(t,T) = I(t)E_r\left\{\exp\left(-\int_t^T r(u)du\right)\middle|\, \mathrm{F}_t\right\} = E_n\left\{\exp\left(-\int_t^T n(u)du\right)I(T)\middle|\, \mathrm{F}_t\right\}} \quad (6)$$

Then Equation (5) becomes:

$$\boldsymbol{ZCIIS(t, T_M, I_0, N) = N\left[\frac{I(t)}{I_0}P_r(t, T_M) - P_n(t, T_M)\right]} \quad (7)$$

Where $\boldsymbol{P_n(t, T_M)}$ is the Zero-coupon bond price at time $\boldsymbol{t}$ for the maturity $\boldsymbol{T}$ in the nominal economy and $\boldsymbol{P_r(t, T_M)}$ is the Zero-coupon bond price at time $\boldsymbol{t}$ for the maturity $\boldsymbol{T}$ in the real economy.

Equation (7) for a valuation at time $t = 0$ simplifies to:

$$\boldsymbol{ZCIIS(0, T_M, N) = N[P_r(0, T_M) - P_n(0, T_M)]} \quad (8)$$

Equations (7) and (8) lead to an important result in the evaluation of the derivative because the pricing formula is independent from the model assumptions given that it follows from the absence of arbitrage. As a result, we are able to unambiguously derive the prices for the zero-coupon bonds starting from the quoted prices of the zero-coupon inflation-indexed swaps (Mercurio, 2005).
In fact, by equating (8) with the actualized nominal value of (1) and obtaining $P_n(0, T_M)$ from the current curve of the nominal zero-coupons, we are able to solve the equation for the unknown quantity $P_r(0, T_M)$.
Therefore, we get:

$$\boldsymbol{P_r(0, T_M) = P_n(0, T_M)[1 + K(T_M)]^M} \quad (9)$$

(Kazziha, 1999) defined the T-forward CPI at time t as the fixed quantity X to be exchanged at time T for the CPI $I(T)$, for which such a swap has a zero value.
From formula (6), we get

$$\boldsymbol{I(t)P_r(t,T) = XP_n(t,T)} \quad (10)$$

The value at time 0 of a T_M-forward CPI can be obtained from the market quotation of $K(T_M)$ applying the formula:

$$\boldsymbol{I_M(0) = I(0) \cdot [1 + K(T_M)]^M} \quad (11)$$

This result is perfectly equivalent to (9).

2.2) Year-on-Year Inflation Indexed Swap (YYIIS) pricing

Pricing a YYIIS is more complicated than the ZCIIS: the pay-off value at time T_i , $t < T_i$ is

$$\boldsymbol{YYIIS(t,T_{i-1},T_i,\psi_i,N) = N\psi_i E_n \left\{\exp\left(-\int_t^{T_i} n(u)du\right)\left[\frac{I(T_i)}{I(T_{i-1})} - 1\right]\middle|\, \mathrm{F}_t\right\}} \quad (12)$$

Assuming $t < T_{i-1}$, we can estimate:

$$\boldsymbol{N\psi_i E_n\{\exp\left(-\int_t^{T_{i-1}} n(u)du\right) E_n\{\exp\left(-\int_{T_{i-1}}^{T_i} n(u)du\right)\left[\frac{I(T_i)}{I(T_{i-1})} - 1\right]\middle|\, F_{t-1}\}|F_t\}} \quad (13)$$

The inner expectation in Equation (13) is the $ZCIIS(T_{i-1},T_i,I(T_{i-1}),1)$:

$$\boldsymbol{N\psi_i E_n\left\{\exp\left(-\int_t^{T_{i-1}} n\, du\right)[P_r(T_{i-1},T_i) - P_n(T_{i-1},T_i)]|\, \mathrm{F}_t\right\} =}$$
$$\boldsymbol{= N\psi_i E_n\left\{\exp\left(-\int_t^{T_{i-1}} n(u)du\right)[P_r(T_{i-1},T_i)]\middle|\, \mathrm{F}_t\right\} - N\psi_i P_n(t,T_i)} \quad (14)$$

The last expected value can be seen as the nominal price of a derivative that pays the price of the zero coupon bond, $P_r(T_{i-1},T_i)$ in nominal unit at time T_{i-1}. If the real rates were deterministic, then this price would be the discounted value, in nominal terms, of the forward price of the real bond. In this case we would have:

$$\boldsymbol{E_n\left\{\exp\left(-\int_t^{T_{i-1}} n(u)du\right)[P_r(T_{i-1},T_i)]\middle|\, \mathrm{F}_t\right\} = P_r(T_{i-1},T_i)P_n(t,T_{i-1}) = \frac{P_r(t,T_i)}{P_r(t,T_{i-1})}P_n(t,T_{i-1})} \quad (15)$$

However, the real rates are stochastic and the expectation is model-dependent. Here, we propose the YYIIS pricing according to the Jarrow-Yildirim (JY) model (Jarrow and Yildirim, 2003).
Denoting by Q_n^T the T-forward nominal measure for a generic maturity T and E_n^T the associated expectation, we can write:

$$\boldsymbol{YYIIS(t,T_{i-1},T_i,\psi_i,N) = N\psi_i P_n(t,T_{i-1})E_n^{T_{i-1}}\{P_r(T_{i-1},T_i)|\mathrm{F}_t\} - N\psi_i P_n(t,T_i)} \quad (16)$$

Recalling the zero-coupon bond price formula in accordance with the Hull-White model (Brigo and Mercurio, 2006):

$$\boldsymbol{P_r(t,T) = A_r(t,T)\exp[-B_r(t,T) \cdot r(t)]} \quad (17)$$

where:

$$\boldsymbol{B_r(t,T) = \frac{1}{a_r}[1 - \exp[-a_r(T - t)]]} \quad (18)$$

$$\boldsymbol{A_r(t,T) = \frac{P_r^M(0,T)}{P_r^M(0,t)}\exp\{B_r(t,T)f_r^M(0,t) - \frac{\sigma_r^2}{4a_r}[1 - \exp(-2a_r t)]B_r(t,T)^2\}} \quad (19)$$

And considering that the instantaneous real rates evolve under $Q_n^{T_i-1}$, according to the stochastic differential equation:

$$\boldsymbol{dr(t) = [-\rho_{n,r}\sigma_n\sigma_r B_n(t,T_{i-1}) + \vartheta_r(t) - \rho_{r,I}\sigma_I\sigma_r - a_r r(t)]dt + \sigma_r dW_r^{T_i-1}(t)} \quad (20)$$

with $W_r^{T_i-1}$ a Brownian motion under the $Q_n^{T_i-1}$ measure, we get:

$$\boldsymbol{YYIIS(t,T_{i-1},T_i,\psi_i,N) = N\psi_i P_n(t,T_{i-1})\frac{P_r(t,T_i)}{P_r(t,T_{i-1})}\exp[C(t,T_{i-1},T_i)] - N\psi_i P_n(t,T_i)} \quad (21)$$

where:

$$\boldsymbol{C(t,T_{i-1},T_i) = \sigma_r B_r(T_{i-1},T_i)[B_r(t,T_{i-1})(\rho_{r,I}\sigma_I - \frac{1}{2}\sigma_r B_r(t,T_{i-1}) + \frac{\rho_{n,r}\sigma_n}{a_n+a_r}(1 + a_r B_n(t,T_{i-1}))) - \frac{\rho_{n,r}\sigma_n}{a_n+a_r}B_n(t,T_{i-1})]} \quad (22)$$

In the Yarrow-Yildirim model (Jarrow and Yildirim, 2003), the expected value for the price of a zero-coupon bond under a nominal forward measure is equal to the current forward price of the bond multiplied by a correction factor, which depends on the instantaneous volatility of the nominal rate, σ_n, of the real rate σ_r, of the CPI σ_I and on the instantaneous correlation between the real rate and the CPI $\rho_{r,I}$.
The exponential of C represents the mentioned correction term: this takes into account the stochasticity of the real rates and, consequently, is zero for $\sigma_r = 0$. The value at time t of the swap inflation-indexed leg is obtained through the summation of all floating payments. Therefore:

$$\boldsymbol{YYIIS(t,\mathcal{T},\Psi,N) = N\psi_{\iota(t)}[\frac{I(t)}{I(T_{\iota(t)-1})}P_r(t,T_{\iota(t)}) - P_n(t,T_{\iota(t)})] + N\sum_{i=\iota(t)+1}^{M}\psi_i\,[P_n(t,T_{i-1})}$$

$$\boldsymbol{\frac{P_r(t,T_i)}{P_r(t,T_{i-1})}\exp[C(t,T_{i-1},T_i)] - P_n(t,T_i)]} \quad (23)$$

Where: $\mathcal{T} \coloneqq \{T_1,\dots,T_M\}$, $\Psi \coloneqq \{\psi_1,\dots,\psi_M\}$ and $\iota(t) = \min\{i: T_i > t\}$ where the first payment after time t has been priced according to (7).

Setting $t = 0$ we get the pricing formula evaluating the leg as of today (Mercurio, 2005):

$$\boldsymbol{YYIIS(0,\mathcal{T},\Psi,N) = N\psi_1[P_r(0,T_1) - P_n(0,T_1)] + N\sum_{i=2}^{M}\psi_i\left[P_n(0,T_{i-1})\frac{P_r(0,T_i)}{P_r(0,T_{i-1})}\exp[C(0,T_{i-1},T_i)] - P_n(0,T_i)\right]} \quad (24)$$

3) CPI index traditional simulation

Through (11), we are able to project the index values in the future according to the swap rates listed on the market following the pricing framework. Since the frequency with which the index is published is monthly, it is necessary to provide a simulation of the CPI with such periodicity (Caligaris and Giribone, 2018). The missing curve points are therefore estimated by adding the logarithm of the monthly increase between a calculated value $\mathfrak{I}_M(0)$ and its subsequent value $\mathfrak{I}_{M+1}(0)$:

$$\boldsymbol{\Delta\mathfrak{I}_M = \frac{ln\left(\frac{\mathfrak{I}_{M+1}(0)}{\mathfrak{I}_M(0)}\right)}{12\cdot\tau}} \quad (25)$$

Where τ is the time interval expressed in year fraction between $\mathfrak{I}_M(0)$ and $\mathfrak{I}_{M+1}(0)$.
The points making up the simulated curve of the consumer price index are defined by the formula:

$$\boldsymbol{\mathfrak{I}_{i+1} = \mathfrak{I}_i\exp(\Delta\mathfrak{I}_M + \mathfrak{R}_M), \mathfrak{I}_M(0) \le \mathfrak{I}_i \le \mathfrak{I}_{M+1}(0)} \quad (26)$$

The standard methodology, suggested by the main benchmark info provider pricing modules, takes into account the index seasonality algebraically adding the normalized residuals $\mathfrak{R}_M$ obtained from the historical values of the CPI, in accordance with the expression (27):

$$\boldsymbol{\mathfrak{R}_M = \frac{\sum_{i=1}^{stagyear}\ln\left[\frac{\mathfrak{I}_{i+1}^{Monthly}}{\mathfrak{I}_i^{Monthly}}\right]}{stagyear} - \frac{\sum_{i=1}^{12\cdot stagyear}\ln\left[\frac{\mathfrak{I}_{i+1}^{Monthly}}{\mathfrak{I}_i^{Monthly}}\right]}{12\cdot stagyear}} \quad (27)$$

Where $\mathfrak{R}_M$ are the standardized residuals obtained from the effect of seasonality over $stagyear$ years. The first contribution is the logarithmic variation of the CPI values on the considered month; the second one represents the overall logarithmic variation recorded in the time period considered for seasonality.

The objective of this study is to propose a deep learning methodology (LSTM network) able to simulate the seasonality of the inflation index. In this way, in addition to introducing a more robust and flexible econometric methodology than the standard one, the integration between the classical quantitative finance theory together with Fintech paradigms can be considered an interesting feature (Bonini *et al.*, 2019).

In fact, the determination of the swap fair value is implemented by applying the formulas described above for the ZCIIS and YYIIS and therefore in total agreement with canonical principles; moreover, a Long Short-Term Memory network will be implemented for a more reliable simulation of the CPI seasonality. The next paragraph deals with the explanation of the architecture for a standard LSTM network.

4) LSTM network architecture and training procedure

LSTM networks are also able to learn long-term relationships between the time intervals of a time series, therefore without the need to pre-set the number of time lags, as occurs in other dynamic recurrent networks, such as Nonlinear AutoRegressive (NAR) and Nonlinear Auto-Regressive with exogenous variables (NARX) (de Simon-Martin *et al.*, 2020).

A common LSTM unit is composed of a cell, an input gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. Intuitively, the cell is responsible for keeping track of the dependencies between the elements in the input sequence. The input gate controls the extent to which a new value flows into the cell, the forget gate controls the extent to which a value remains in the cell and the output gate controls the extent to which the value in the cell is used to compute the output activation of the LSTM unit (Hochreiter and Schmidhuber, 1997). The activation function of the LSTM gates is often the logistic sigmoid. Figure 1 shows how the flux of a data sequence Y with C features (or channels) of length S has been processed into a LSTM layer. In the block diagram, h_t and c_t are, respectively, the output (also known as hidden state) and the cell state at time t.

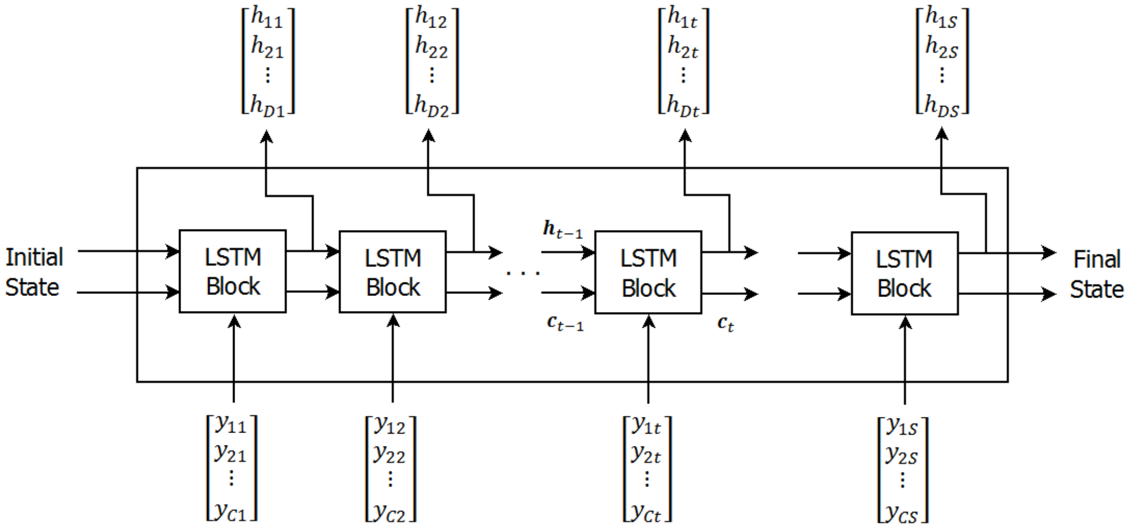
$\begin{bmatrix} h_{11} \\ h_{21} \\ \vdots \\ h_{D1} \end{bmatrix}$
$\begin{bmatrix} h_{12} \\ h_{22} \\ \vdots \\ h_{D2} \end{bmatrix}$
$\begin{bmatrix} h_{1t} \\ h_{2t} \\ \vdots \\ h_{Dt} \end{bmatrix}$
$\begin{bmatrix} h_{1S} \\ h_{2S} \\ \vdots \\ h_{DS} \end{bmatrix}$
Initial State
LSTM Block
LSTM Block
h_{t-1}
c_{t-1}
LSTM Block
c_t
LSTM Block
Final State
$\begin{bmatrix} y_{11} \\ y_{21} \\ \vdots \\ y_{C1} \end{bmatrix}$
$\begin{bmatrix} y_{12} \\ y_{22} \\ \vdots \\ y_{C2} \end{bmatrix}$
$\begin{bmatrix} y_{1t} \\ y_{2t} \\ \vdots \\ y_{Ct} \end{bmatrix}$
$\begin{bmatrix} y_{1S} \\ y_{2S} \\ \vdots \\ y_{CS} \end{bmatrix}$

Figure 1. LSTM network architecture

The first LSTM block uses the initial state of the network and the first time-step of the sequence in order to compute the first output and the first update of the cell state. At time t, the block uses the current state of the network (c_{t-1}, h_{t-1}) and the next step of the sequence for estimating the output and updating the current state of the cell c_t. The layer state is characterized by the hidden state (also known as the output state) and the cell state. The hidden state at time step t contains the output of the LSTM layer for the current time step. The cell state contains the information learnt in the previous steps. For each time step, the layer adds or removes information from the cell state. The layer controls these updates using gates. The following components control the cell state and the hidden state of the layer (Hochreiter and Schmidhuber, 1997):

- Input gate (i): Control level of cell state update
- Forget gate (f): Control level of cell state reset (forget)
- Cell candidate (g): Add information to cell state
- Output gate (o): Control level of cell state added to hidden state

Fig. 2 shows how the gates (i, f, g, o) process the signal at time t

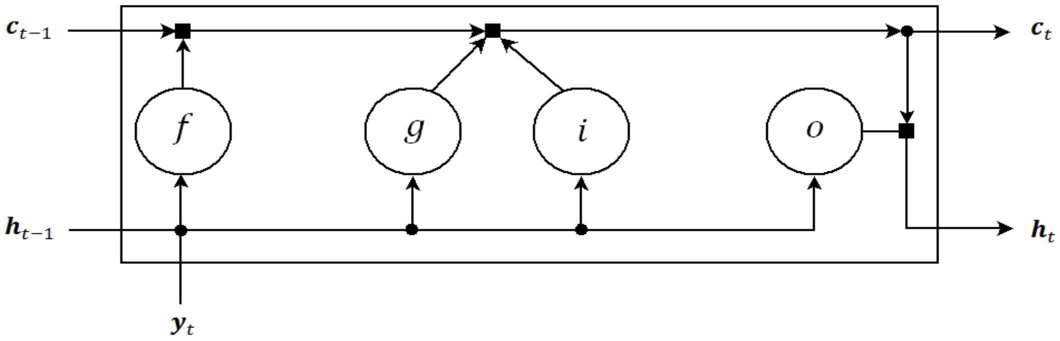

Figure 2. Signal processed by the gates

In a LSTM, the parameters that are subjected to calibration are: the input weights (W), the recurrent weights (R) and the biases (b) (de Simon-Martin *et al.*, 2020). W, R and b are the arrays built through the concatenations of such parameters for each component: $W = \left(W_i, W_f, W_g, W_o\right)^{\top} R = \left(R_i, R_f, R_g, R_o\right)^{\top} b = \left(b_i, b_f, b_g, b_o\right)^{\top}$
where i, f, g and o denote the input gate, the forget gate, the cell candidate and the output gate, respectively.
At time step t, the cell state is given by:

$$\boldsymbol{c_t = f_t \odot c_{t-1} + i_t \odot g_t} \quad (28)$$

Where $\odot$ is the Hadamard product operator.
At time step t, the hidden state is given by:

$$\boldsymbol{h_t = o_t \odot \sigma_c(c_t)} \quad (29)$$

Where σ_c is the activation function of the state (typically a hyperbolic tangent).

The following equations define the components at time step t:

- Input gate (i): $\boldsymbol{i_t = \sigma_g\left(W_i y_t + R_i h_{t-1} + b_i\right)}$ (30)

- Forget gate (f): $\boldsymbol{f_t = \sigma_g\left(W_f y_t + R_f h_{t-1} + b_f\right)}$ (31)

- Cell candidate (g): $\boldsymbol{g_t = \sigma_c\left(W_g y_t + R_g h_{t-1} + b_g\right)}$ (32)

- Output gate (o): $\boldsymbol{o_t = \sigma_g\left(W_o y_t + R_o h_{t-1} + b_o\right)}$ (33)

σ_g is the activation function of the gate, which is typically a sigmoid.

LSTMs are supervised networks, as a result, after the design of the model, it is essential to implement a robust algorithm for the training phase. This is the part in which the designer decides how many neurons must be implemented in order to make reliable predictions. In order to obtain valid models for forecasting purposes it is necessary to conduct statistical and econometric tests. The objective of the first kind of test is to tune the LSTM in order to have a good fitting of the training dataset.
The gap between the target and the model output is reduced through an ADAM optimizer as the network training process progresses, so it may happen that the estimated relationship returns a perfect fit of the sampled data (in-sample), making vain the attempt at generalization, fundamental for making the network capable of processing different data (out-of-sample).
For this reason and especially in the field of deep learning where there is a huge number of parameters to tune in order to capture highly non-linear relationships, special measures for avoiding overfitting must be taken into consideration. As a result, the first intervention, shared also with traditional recurrent networks, such as NAR and NARX, is to work directly on the dataset through a random-splitting method. The data set configuration used for the network is:

- 70% of the set will form the training set, thus the optimization will be carried out with respect to its loss function (J) only.
- 15% of the set will be assigned to the validation set, thus, despite the weights being updated with respect to the train set, the algorithm saves the weights that minimize J on the validation set, in order to avoid data overfitting and trying to reach a good generalization.
- 15% of the data set will form the test set, so that the network performance can be measured on data that it has never seen before, as the ultimate objective of a neural network user is to employ the network on completely new data.

The second kind of statistical measures, which are traditionally applied in the field of deep learning, work directly on the network. The implemented measures can be summarized as follows:

- Adding a term to the traditional loss function ($RMSE$) which put in a penalty (the λ coefficient) if a further weight (ω) associated to an arch has been activated:

  $$\boldsymbol{J = RMSE + \frac{1}{2}\lambda\|\omega\|^2} \quad (34)$$

- Dropout, which is a technique consisting of training only a group of randomly selected neurons rather than the entire network: a percentage (a popular choice is 25%) determines how many neurons to choose and the remaining ones are deactivated. Since the neurons and the relative weights are continuously modified, it is thus possible to avoid overfitting.

These metrics are thus implemented in the forecaster in order to have a reliable fitting.
Given that the objective is to perform a prediction of the most reasonable CPI projections, we also implement a test which has an econometric nature. It is based on the verification of the autocorrelation error absence so that the model error is unstructured and the predicted values can be econometrically reliable.

5) Comparison between standard and LSTM techniques for the CPI projection

In order to compare the standard inflation bootstrap methodology with the LSTM approach, we use the market data retrieved from Bloomberg on 31st December 2020. Swap rate values, $K(T_M)$, quoted by the market at the reference date are reported in Table 1, together with the estimation of the CPI projections, $\mathfrak{I}_M(0)$ and the $\Delta\mathfrak{I}_M$, according to Formula (11) and (25). The estimation of $\Delta\mathfrak{I}_M$ is useful in order to have the inflation values expressed on monthly basis (Giribone, 2020). According to Equation (26), this information allows us to project the CPI values for the next 10 years using a market-oriented approach without taking into account the seasonality. In order to add this essential contribution for the forecast into the model, we have to consider the monthly normalized residuals, $\mathfrak{R}_M$, calculated starting from the past CPI realizations. The traditional way to implement this task is to apply Equation (27). Using the traditional market standard preference to consider the previous five years of the CPI time-series, we get the $\mathfrak{R}_M$ reported in the second column of Table 2. Applying recursively equation (26), the projections for the CPI are obtained for the following years. These simulations are reported in Fig. 3 together with the past values. The black line represents the past five years CPI values used for the estimation of the seasonality effect; the blue line represents the CPI projections without seasonality: it connects the blue dots which are the $\mathfrak{I}_M(0)$ whose estimations are strictly connected to the $K(T_M)$ quotation and the red line represents the projections of the CPI index taking into account the seasonality through the monthly annualized residuals $\mathfrak{R}_M$. According to this approach, we have a “ready-to-use” seasonality model that is able to take into consideration market-implied IIS rates. Despite this being the market-standard approach proposed by the main info-providers, this methodology has important drawbacks from an econometric perspective.

In fact, it does not take into account basic indicators, such as R^2 or the absence of autocorrelation in the errors: it substantially repeats the same twelve-monthly residuals over time, as shown in Fig. 4.
The black line represents the historical CPI log-returns and the red one shows their projections in accordance with the standard approach.

T_M	$K(T_M)$ [%]	$\mathfrak{I}_M(0)$	$\Delta\mathfrak{I}_M$ [%]
1	1.206	105.8010	0.0999
2	1.010	106.6624	0.0676
3	0.966	107.5989	0.0729
4	0.972	108.6642	0.0821
5	1.003	109.8862	0.0932
6	1.034	111.1986	0.0989
7	1.055	112.5090	0.0976
8	1.081	113.9302	0.1046
9	1.111	115.4697	0.1119
10	1.133	117.0034	0.1100

Table 1. Mid Price $\boldsymbol{K(T_M)}$,$\boldsymbol{\mathfrak{I}_M(0)}$ and $\boldsymbol{\Delta\mathfrak{I}_M}$ (Market Data: 31st December 2020)

Month (M)	Standard Approach [%]	LSTM-Method I [%]
January (1)	-1.198	-0.761
February (2)	0.174	-0.018
March (3)	0.836	0.575
April (4)	0.264	0.292
May (5)	0.081	0.019
June (6)	0.086	0.009
July (7)	-0.529	-0.272
August (8)	-0.027	-0.375
September (9)	0.246	0.339
October (10)	0.111	0.188
November (11)	-0.219	-0.105
December (12)	0.175	0.108

Table 2. Historical Normalized residuals, $\boldsymbol{\mathfrak{R}_M}$, in accordance with the standard methodology (Market Data: 12/31/2020)

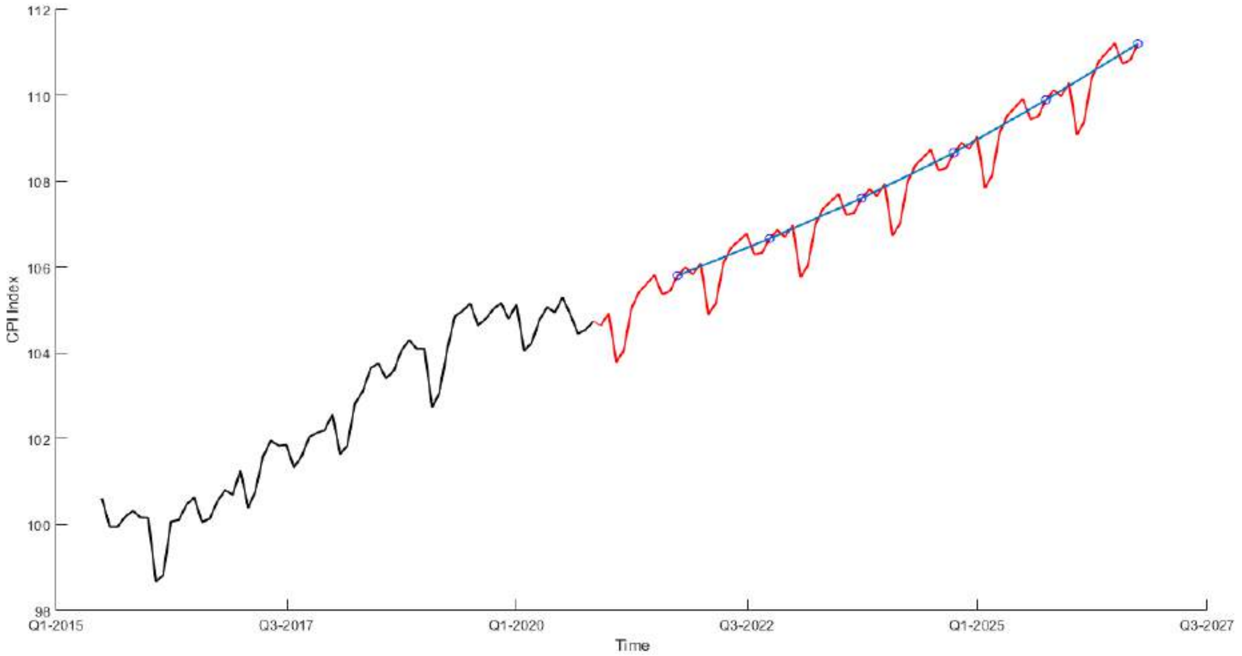

Figure 3. CPI time series and its projection (standard methodology)

The idea is to improve the seasonality model using a method with the following characteristics:
A) it is reliable from an econometric and statistical perspective
B) it is able to deal with potentially high non-linearities recorded by the financial series.
C) it takes into account market information (ZCIIS rates) and, consequently, it is aligned with traditional quantitative finance market best-practice (see paragraph 2).

With these objectives in mind, we design a Long Short-Term Memory (LSTM) network with the characteristics illustrated in paragraph 4. For the training set, we use the monthly return of the index computed in the last 5 years: $\ln\left[\frac{\Im_{i+1}^{Monthly}}{\Im_{i}^{Monthly}}\right]$. The number of hidden units in the LSTM block is tuned in function of the performances recorded by the network. Using a layer made of 100 neurons, adopting an ADAM optimizer and implementing a drop-out technique in order to avoid overfitting we can achieve excellent results in the training phase (de Simon-Martin et al., 2020). From a statistical point of view, as shown in Fig. 5, we obtain a high R^2, meaning that the fitting over the historical time series can be considered extremely good. From an econometric point of view, Fig. 6 shows that the auto-correlation in the errors for the tuned model has been kept, with a confidence interval equal to 95%, under an acceptable threshold (represented in red dotted lines) for the non-zero lags (Tsay, 2010).

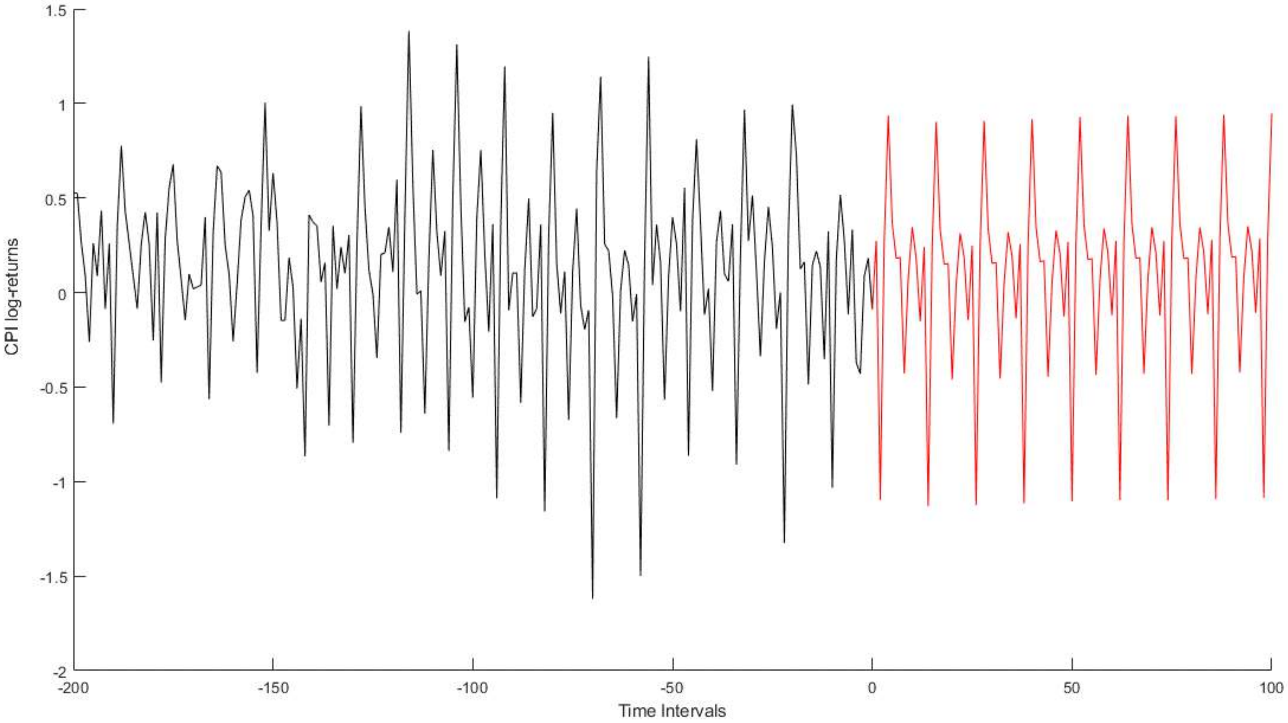

Figure 4. Historical and perspective estimation for seasonality using standard methodology

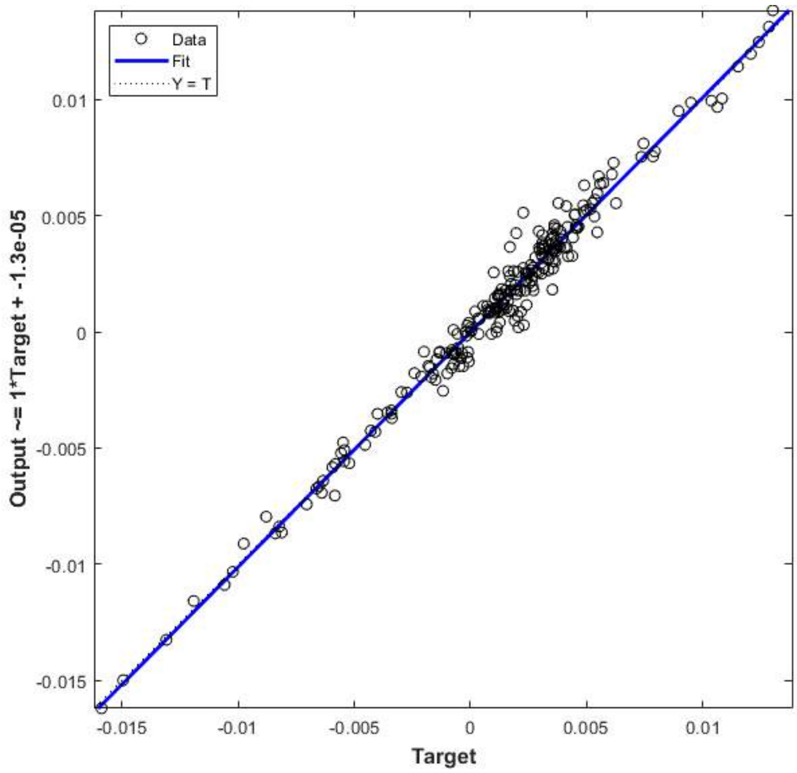

Figure 5. Regression plot for the LSTM network. $\boldsymbol{R^2} = \mathbf{0.98}$

Having checked the reliability of the LSTM network through statistical and econometric test, we proceed to make the CPI forecasts. Facing the forecasting problem with the Deep Learning approach, the projections have a more realistic forward-looking behavior thanks to both the advanced technology (deep learning) and the careful tuning as shown in Figure 7.

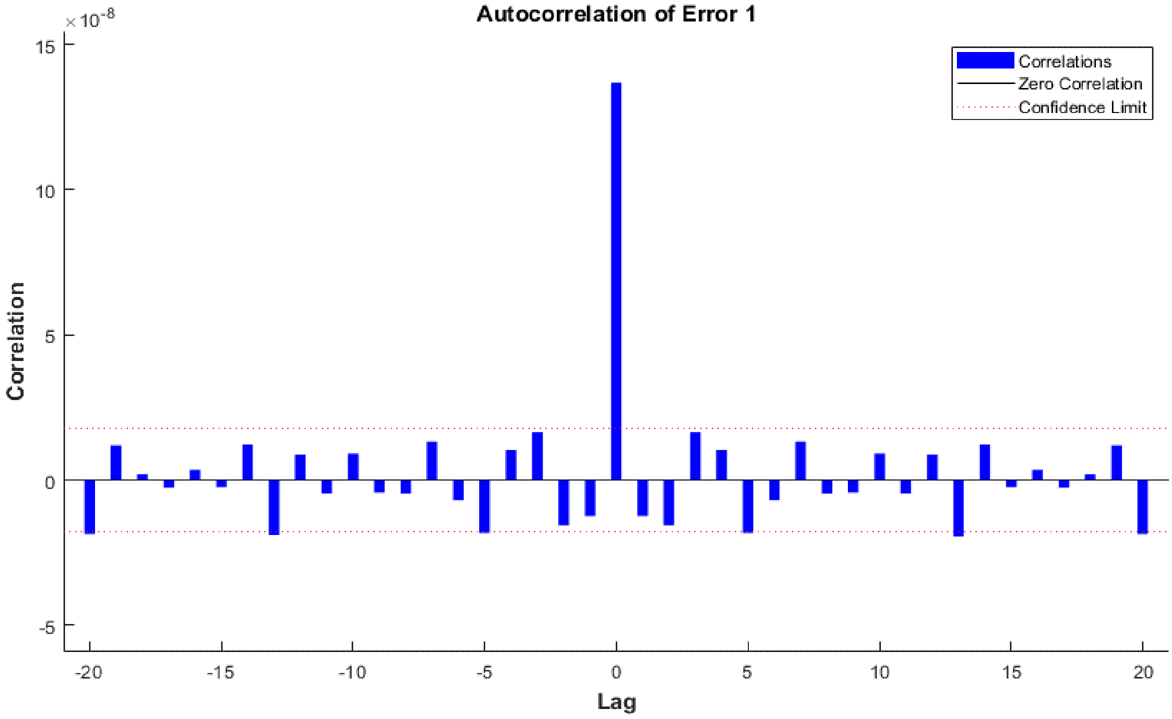

Figure 6. Auto-correlation error for LSTM network

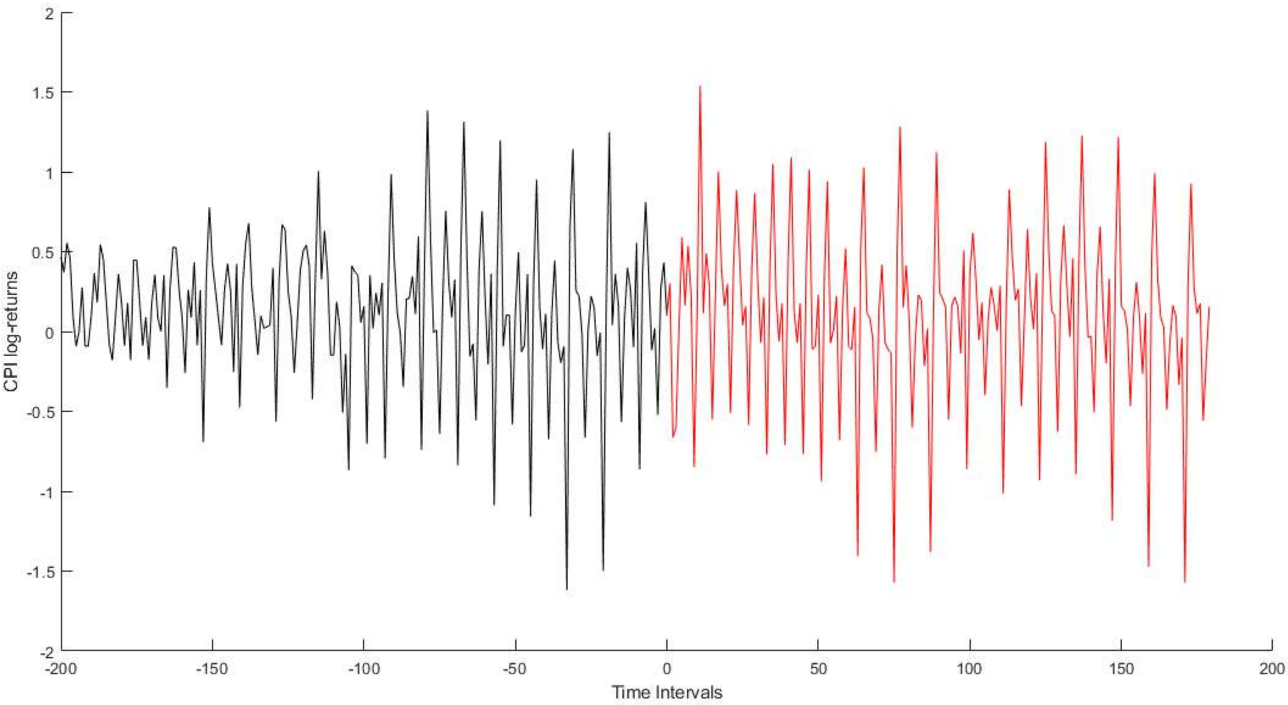

Figure 7. Historical and perspective estimation for seasonality using the Deep Learning methodology

Performing these seasonality projections guarantees to be compliant with points A) and B) of the desired requirements but not with the last one. In fact the log-returns projections cannot be used as-is in the pricing framework because they do not pass through the market-implied future values (see Table 1). Starting from these LSTM projections, two approaches can be followed in order to take into consideration this quantitative finance aspect:
LSTM-Method I) determines the monthly standardized residuals along the overall forecasted values.
LSTM-Method II) determines the monthly standardized residuals among the time intervals delimited by the market-implied rates.
LSTM-Method I) does not introduce an innovative idea with respect to the standard method with the difference that this is applied to the forecasted CPI values instead of using the historical values. It is substantially a forward-looking standard method. The twelve standardized residuals are reported in the third column of Table 2.
LSTM-Method II) applies the normalization of future residuals not on the entire forecasted horizon, but it considers the projected values for the computation among the time intervals delimited by the market-implied rates. This approach is less similar to the traditional one but it allows a more precise CPI seasonality projection in which the constrained C) remains satisfied. For this reason, we believe that the second approach has to be preferred.
LSTM-Method I) does not change the iterative formula (26) for $\mathfrak{I}$ because the seasonality term remains constant over the years and it can assume only twelve values, as reported in Table 2. The main difference between the standard methodology and this approach is a different estimation of the normalized residuals: the former considers the previous five years of the CPI values (as a result it is a backward-looking method), while the latter improves the forecasting, considering the projected five years values (as a result it is a forward-looking method). Despite this improvement, it remains a static technique.
LSTM-Method II) does not only introduce a forward-looking view, but also adds a more interesting contribution in terms of dynamics. In this second case the formula (26) slightly changes, because of the insertion of an index j that allows to clarify the exact contribution of $\boldsymbol{\mathfrak{R}_M^j}$ in connection with the cluster it belongs to. This happens because it changes between the time intervals defined by the listed ZCIIS ($\boldsymbol{j = 1, \dots, T_M}$).

6) Robust check of the LSTM results

One of the main problems associated with dynamics neural networks for forecasting is that they are able to provide only projections without their related confidence intervals. This is mainly due to the fact that they are able to learn highly non-linear predictions in a time series without the need to make any assumptions regarding statistical distributions.

This is certainly a strength for capturing non-standard relationships, but it does not allow the analyst to express the results in terms of confidence bands. In order to improve the readability of the LSTM results and, consequently, help the user to be confident on the projection made by the network, we design a traditional econometric Seasonality ARIMA model (SARIMA). The model is an ARIMA(p,d,q) error model seasonally integrated with Seasonal MA(12). We maintain the seasonality effect equal to one year, that is 12 lags and we implement the traditional Gaussian distribution.

The main idea is to be able to calculate the confidence intervals for the econometric models at 5% and 95% using percentiles method in a Monte Carlo method based on the SARIMA and comparing these extreme values with the LSTM point predictions. This comparison can help the analyst to understand if the simulated seasonality can be reasonable. This comparisons between innovative (and maybe black-box) methodologies and traditional statistical models are more and more widespread in the Explainable Artificial Intelligence.

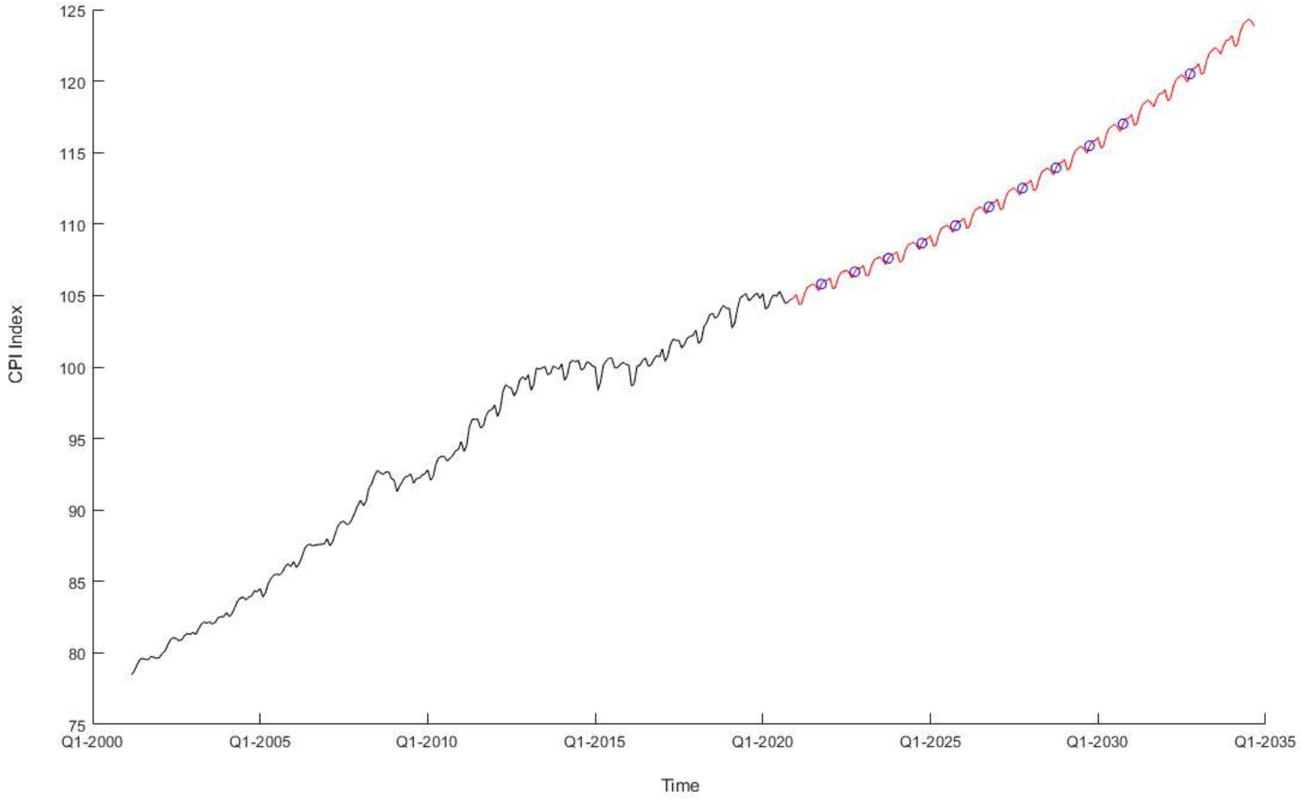

Figure 8. CPI time series and its projection (LSTM-Method I)

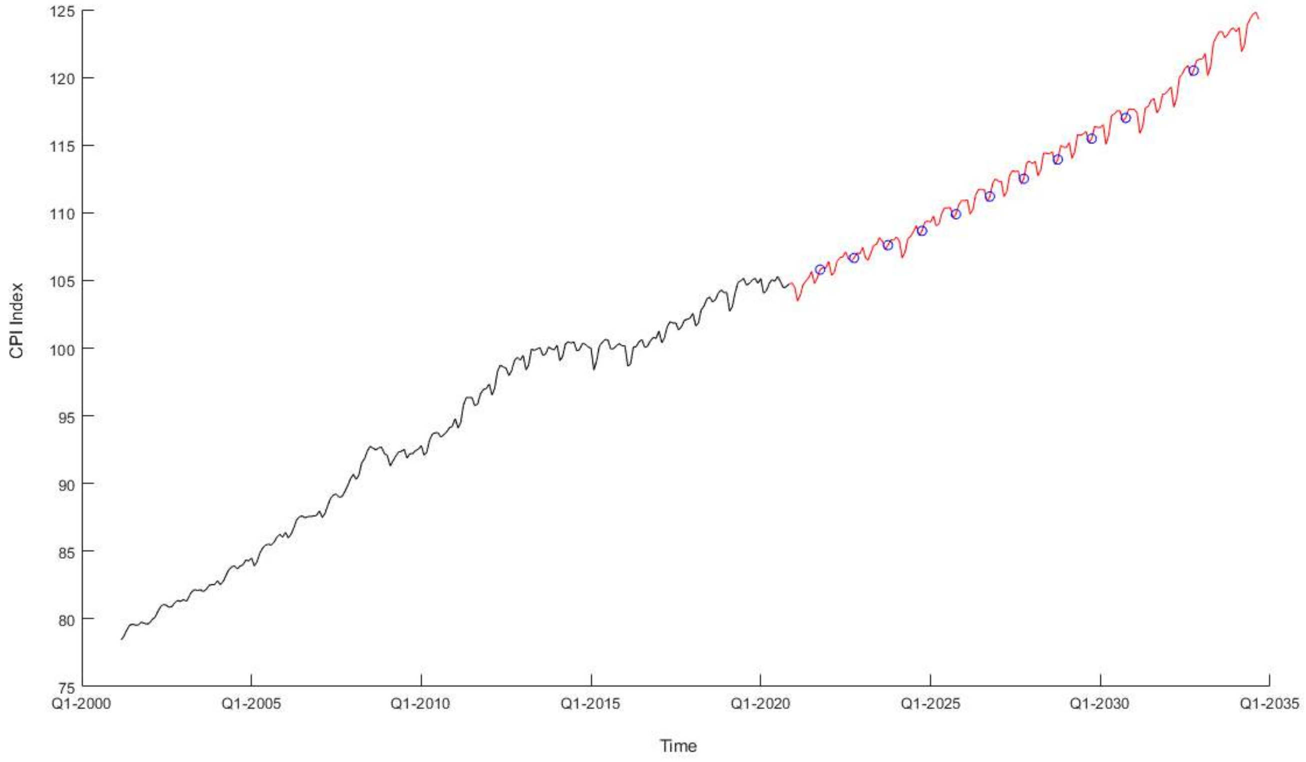

Figure 9. CPI time series and its projection (LSTM-Method II)

The first step is to choose the order for our Seasonal ARIMA(p,d,q) model that is find the best p, d and q parameters. With this aim we generate two tensors in which we store the Akaike's Information Criterion (AIC) and the Bayesian Information Criteria (BIC) estimated iteratively for all possible sets of parameters: $p = 1, \dots, 5$, $d = 1, \dots, 5$ and $q = 1, \dots, 5$. The minimum AIC and BIC is for a SARIMA(1,1,2). The second step is to find the estimation of the model and then perform 100,000 simulations for the next months (i.e.120) for computing the 5% and 95% in correspondence of these time steps. Figure 10 highlights that the empirical confidence intervals of the traditional econometric SARIMA model (red lines) include almost all the LSTM network predictions (blue lines). This information can help the analyst to understand that the Machine Learning produce reasonable predictions because its outputs stay inside the extreme forecasts done with a methodology for which confidence bands can be estimated.

This helps to reduce the black-box effect and consequently to better explain the outcomes of the deep learning methodology.

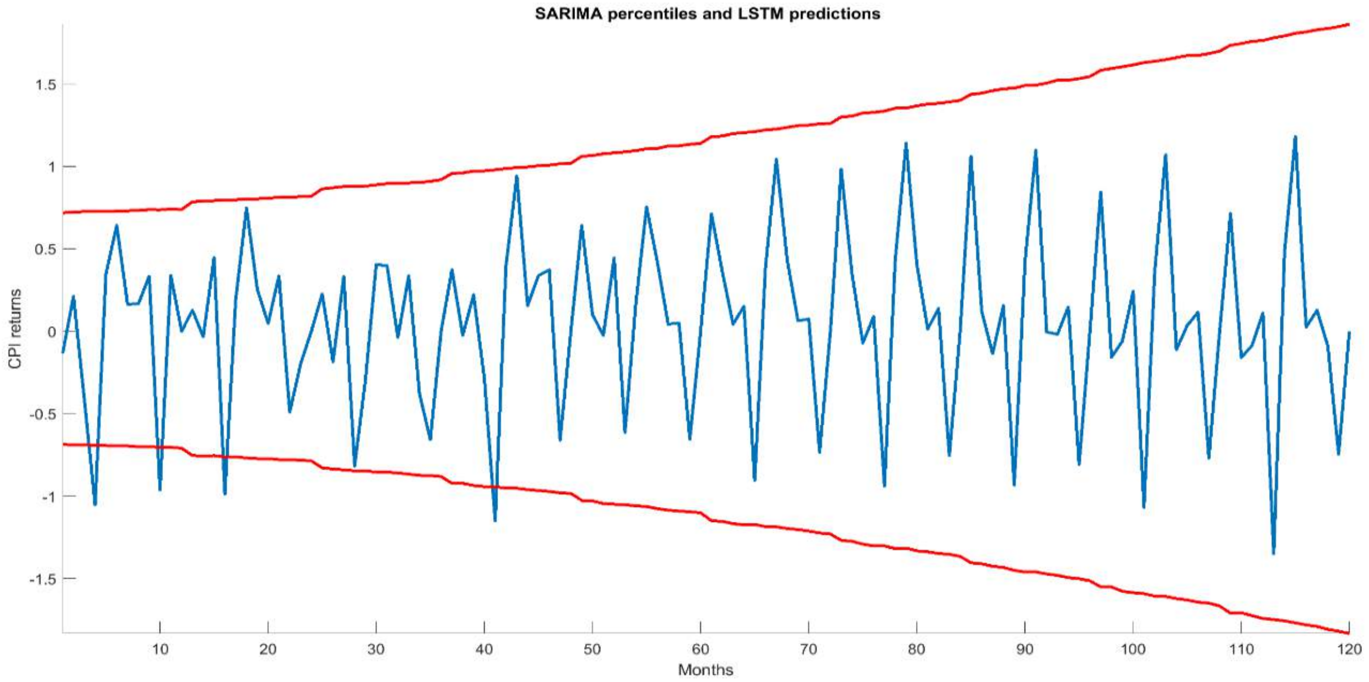

Figure 10. SARIMA(1,1,2) with Seasonal lag equal to 12 percentiles and the LSTM projection

7) Market Case: YYIIS pricing

The seasonality model obviously has an impact on the derivative fair-value that is not always negligible.

In this paragraph, we proceed with the valorization of a YYIIS swap using the two approaches described in the previous paragraphs. The main financial characteristics are reported in Table 3.

The valuation date of the "In Arrears" swap is 31st December 2020, thus we can use the historical and prospective inflation data already computed in the previous paragraphs. Regarding the discount curve we use, according to the new benchmark standard for collateralized derivatives is the EUR OIS ESTR term structure.

As a result, zero rates and discount factors used for pricing are those implied from the new market benchmark curve (see Table 4).

YOY Swap	Receiving Leg	Paying Leg
Leg Type	Y-o-Y Inflation	Fixed
Notional	10 MM	10 MM
Currency	Euro	Euro
Index	CPTFEMU Index	Fixed Coupon: 2.1%
Effective Date	31st Dec. 2010	31st Dec. 2010
Maturity Date	31st Dec. 2030	31st Dec. 2030
Lag	3 Month	-
Interpolation	Monthly	-
Spread	0	-
Reset Frequency	Semi-Annual	-
Payment Freq.	Semi-Annual	Annual
Day Count	ACT/360	ACT/ACT
Discount Curve	EUR-OIS-ESTR	EUR-OIS-ESTR

Table 3. Year-on-Year Inflation Indexed Swap Financial Characteristics

Term	Market Rates	Zero Rates	Discount Factors
1 DY	-0.583	-0.5911	1.000016
1 WK	-0.555	-0.56274	1.000108
2 WK	-0.555	-0.56277	1.000216
1 MO	-0.5245	-0.5319	1.000452
2 MO	-0.56	-0.56804	1.000919
3 MO	-0.564	-0.57224	1.001428
4 MO	-0.5685	-0.57694	1.001899
5 MO	-0.5703	-0.57892	1.00243
6 MO	-0.574	-0.58281	1.002894
7 MO	-0.5778	-0.58682	1.003414
8 MO	-0.581	-0.59023	1.003953
9 MO	-0.586	-0.59546	1.004464
10 MO	-0.5875	-0.59714	1.004986
11 MO	-0.5895	-0.59933	1.005516
12 MO	-0.592	-0.60203	1.006038
18 MO	-0.602	-0.61189	1.009195
2 YR	-0.61262	-0.61446	1.012365
3 YR	-0.60932	-0.61118	1.018504
4 YR	-0.60118	-0.60266	1.024433
5 YR	-0.58329	-0.58494	1.029695
6 YR	-0.55663	-0.55851	1.034094
7 YR	-0.5222	-0.52437	1.037402
8 YR	-0.48456	-0.48686	1.039745
9 YR	-0.44388	-0.4465	1.041054
10 YR	-0.39831	-0.40124	1.040974

Table 4. EUR-OIS-ESTR Discount Curve. Reference Date: 31st December 2020. Source: Bloomberg®

Using the pricing formulas derived in paragraph 2, we proceed with the estimation of the future cash-flows for the swap and then we go through the discounting process for obtaining the NPVs for the two legs. The difference between the two NPVs gives the price of the swap. In detail:

- the discounted Cash Flows for the fixed paying leg of the swap are equal to -2,159,760.13 Euro (see Table 5).
- the discounted Cash Flows for the inflation-indexed receiving leg of the swap using the standard seasonality approach are equal to +1,178,818.67 Euro (see Table 6).
- the discounted Cash Flows for the inflation-indexed receiving leg of the swap using the deep learning architecture are equal to +1,231,750.60 Euro (see Table 7).

Accrual Start	Payment Date	Days	Notional	Coupon	Payment	Discount	Zero Rate	PV
12/31/2020	12/31/2021	365	-10000000	2.1	-209998.43	1.006032	-0.60144	-211265.24
12/31/2021	12/30/2022	364	-10000000	2.1	-209424.66	1.012343	-0.61422	-212009.62
12/30/2022	12/29/2023	364	-10000000	2.1	-209424.66	1.018469	-0.61115	-213292.62
12/29/2023	12/31/2024	368	-10000000	2.1	-211152.26	1.02442	-0.60276	-216308.66
12/31/2024	12/31/2025	365	-10000000	2.1	-209998.43	1.029709	-0.5852	-216237.26
12/31/2025	12/31/2026	365	-10000000	2.1	-210000.00	1.034121	-0.55895	-217165.47
12/31/2026	12/31/2027	365	-10000000	2.1	-210000.00	1.037446	-0.52497	-217863.67
12/31/2027	12/29/2028	364	-10000000	2.1	-208854.03	1.039792	-0.48775	-217164.65
12/29/2028	12/31/2029	367	-10000000	2.1	-211145.97	1.041122	-0.44749	-219828.70
12/31/2029	12/31/2030	365	-10000000	2.1	-210000.00	1.041068	-0.40225	-218624.24

Table 5. Fixed Payment Leg Valuation

Accrual Start	Accrual End	Days	Notional	Reset Date	Reset Rate	Reset Price	Coupon	Payment	Discount	Zero Rate	PV
12/31/2020	06/30/2021	181	10000000	03/01/2021	0.6337	104.86851	0.6337	31860.78	1.002892	-0.582367	31952.92
06/30/2021	12/31/2021	184	10000000	09/01/2021	1.76383	**105.80096**	1.76383	90151.2	1.006032	-0.601438	90695.04
12/31/2021	06/30/2022	181	10000000	03/01/2022	0.52997	106.07902	0.52997	26645.85	1.009188	-0.611429	26890.68
06/30/2022	12/30/2022	183	10000000	09/01/2022	1.09685	**106.66237**	1.09685	55756.46	1.012343	-0.614222	56444.67
12/30/2022	06/30/2023	182	10000000	03/01/2023	0.59075	106.97656	0.59075	29865.85	1.015412	-0.612777	30326.13
06/30/2023	12/29/2023	182	10000000	09/01/2023	1.16675	**107.59893**	1.16675	58985.67	1.018469	-0.611149	60075.11
12/29/2023	06/28/2024	182	10000000	03/01/2024	0.70232	107.97574	0.70233	35506.43	1.021433	-0.607088	36267.44
06/28/2024	12/31/2024	186	10000000	09/01/2024	1.25114	**108.66416**	1.25114	64642.41	1.02442	-0.60276	66221
12/31/2024	06/30/2025	181	10000000	03/01/2025	0.84101	109.11734	0.84101	42283.91	1.02709	-0.59417	43429.38
06/30/2025	12/31/2025	184	10000000	09/01/2025	1.39773	**109.88619**	1.39773	71439.3	1.029709	-0.585203	73561.69
12/31/2025	06/30/2026	181	10000000	03/01/2026	0.90813	110.38104	0.90813	45658.9	1.031968	-0.572289	47118.54
06/30/2026	12/31/2026	184	10000000	09/01/2026	1.46395	**111.19565**	1.46395	74823.95	1.034121	-0.558946	77377.04
12/31/2026	06/30/2027	181	10000000	03/01/2027	0.89757	111.69057	0.89757	45127.69	1.035863	-0.542196	46746.12
06/30/2027	12/31/2027	184	10000000	09/01/2027	1.45352	**112.50897**	1.45352	74291.22	1.037446	-0.524965	77073.13
12/31/2027	06/30/2028	182	10000000	03/01/2028	0.97434	113.05558	0.97434	49258.37	1.038719	-0.50642	51165.62
06/30/2028	12/29/2028	182	10000000	09/01/2028	1.55144	**113.93017**	1.55144	78433.73	1.039792	-0.487753	81554.73
12/29/2028	06/29/2029	182	10000000	03/01/2029	1.06202	114.53349	1.06202	53690.98	1.040562	-0.467854	55868.81
06/29/2029	12/31/2029	185	10000000	09/01/2029	1.61278	**115.46973**	1.61278	82879.04	1.041122	-0.447493	86287.18
12/31/2029	06/28/2030	179	10000000	03/01/2030	1.05645	116.06797	1.05645	52528.81	1.041221	-0.425386	54694.11
06/28/2030	12/31/2030	186	10000000	09/01/2030	1.58155	**117.00341**	1.58155	81713.54	1.041068	-0.402249	85069.33

Table 6. Inflation-Indexed Receiving Leg Valuation using the standard approach

Accrual Start	Accrual End	Days	Notional	Reset Date	Reset Rate	Reset Price	Coupon	Payment	Discount	Zero Rate	PV
12/31/2020	06/30/2021	181	10000000	03/01/2021	0.28164	104.68732	0.28164	14160.14	1.002892	-0.582367	14201.09
06/30/2021	12/31/2021	184	10000000	09/01/2021	2.11633	**105.80096**	2.11633	108167.9	1.006032	-0.601438	108820.38
12/31/2021	06/30/2022	181	10000000	03/01/2022	1.12420	106.39734	1.12420	56522.2	1.009188	-0.611429	57041.53
06/30/2022	12/30/2022	183	10000000	09/01/2022	0.49757	**106.66237**	0.49757	25293.1	1.012343	-0.614222	25605.29
12/30/2022	06/30/2023	182	10000000	03/01/2023	0.71823	107.04610	0.71823	36310.58	1.015412	-0.612777	36870.20
06/30/2023	12/29/2023	182	10000000	09/01/2023	1.03022	**107.59893**	1.03022	52083.28	1.018469	-0.611149	53045.21
12/29/2023	06/28/2024	182	10000000	03/01/2024	-0.95952	107.08395	-0.95952	-48509.1	1.021433	-0.607088	0.00
06/28/2024	12/31/2024	186	10000000	09/01/2024	2.92979	**108.66416**	2.92979	151372.4	1.02442	-0.60276	155068.92
12/31/2024	06/30/2025	181	10000000	03/01/2025	0.91998	109.16515	0.91998	46254.4	1.02709	-0.59417	47507.44
06/30/2025	12/31/2025	184	10000000	09/01/2025	1.31665	**109.88619**	1.31665	67295.66	1.029709	-0.585203	69294.94
12/31/2025	06/30/2026	181	10000000	03/01/2026	0.65143	110.24469	0.65143	32752.3	1.031968	-0.572289	33799.33
06/30/2026	12/31/2026	184	10000000	09/01/2026	1.72312	**111.19862**	1.72312	88070.79	1.034121	-0.558946	91075.85
12/31/2026	06/30/2027	181	10000000	03/01/2027	0.66842	111.57088	0.66842	33606.8	1.035863	-0.542196	34812.04
06/30/2027	12/31/2027	184	10000000	09/01/2027	1.67458	**112.50897**	1.67458	85589.4	1.037446	-0.524965	88794.38
12/31/2027	06/30/2028	182	10000000	03/01/2028	1.19056	113.18071	1.19056	60189.29	1.038719	-0.50642	62519.76
06/30/2028	12/29/2028	182	10000000	09/01/2028	1.31999	**113.93017**	1.31999	66733.04	1.039792	-0.487753	69388.48
12/29/2028	06/29/2029	182	10000000	03/01/2029	0.97715	114.48816	0.97715	49400.22	1.040562	-0.467854	51403.99
06/29/2029	12/31/2029	185	10000000	09/01/2029	1.70740	**115.46973**	1.70740	87741.21	1.041122	-0.447493	91349.31
12/31/2029	06/28/2030	179	10000000	03/01/2030	0.39262	115.69663	0.39262	19521.73	1.041221	-0.425386	20326.44
06/28/2030	12/31/2030	186	10000000	09/01/2030	2.24632	**117.00341**	2.24632	116059.7	1.041068	-0.402249	120826.04

Table 7. Inflation-Indexed Receiving Leg Valuation using the LSTM approach

The gap between the values from the two pricing methodologies is equal to -52,931.93 and, consequently, the percentage error, measured as the ratio between the absolute evaluation discrepancy and the notional of the derivative, is higher than 0.529%.

It is worth noting that the simulated future CPIs implied by the rates of the listed ZCIIS are exactly the same independently of the implemented seasonality method (i.e. standard or LSTM-Method 2). These values are highlighted in bold in Tables 6 and 7.

LSTM-Method I) provides a result very close to the standard methodology: the NPV of the inflation-indexed receiving leg is 1,175,808.67 Euro.

8) Conclusions

This study shows how a Deep Learning methodology can be usefully implemented in a pricing framework, aiming at determining the fair value of derivatives linked to the inflation index.

The Long Short-Term Memory Network allows to identify the effect of seasonality in a more reliable way compared to traditional methodologies. In fact, the proposed technique is able to simulate the future values of the time series by applying the described rigorous statistical and econometric tests, reasonably guaranteeing the reliability of the forecasts.

On the contrary, the traditional approach, based on the estimation of the historical normalized residuals, does not consider these important tests and it is not able to capture highly nonlinear relationships as a LSTM network does. It is particularly interesting considering how artificial intelligence paradigms can be integrated with traditional pricing methodologies in the field of quantitative finance.

In summary, the study shows that seasonality has larger impacts than previously expected on the Inflation-Indexed Swaps valuation, especially when counterparties exchange a fixed interest rate compared to a floating rate.

The proposed methodology also combines market elements with a machine learning approach, making the method more dynamic, despite inflation rates being estimated only periodically.

In addition, thanks to this dynamic approach, the model proposed allows financial institutions to better estimate future cash flows that counterparties have to exchange over the years, so to make the risk management process more accurate compared to more traditional approaches.

Despite the results being interesting, this research represents only a preliminary study in this area and further analyses to test and to improve the model are thus required.

Possible future researches could aim either at determining which factors impact the most on the variability of the results, or at seeing the implications of such methodology when applied to derivative contracts, written on underlyings (such as commodity and energy derivatives) where the seasonality effect is of fundamental importance.

Bibliography

- Bonini S., Caivano G., Cerchiello P., Giribone P. G. (2019). Artificial Intelligence: Applications of Machine Learning and Predictive Analytics in Risk Management. AIFIRM (Italian Association of Financial Industry Risk Managers) position paper, N. 14, pp. 1-164
- Brigo D., Mercurio F. (2006). Interest Rate Models: Theory and Practice with smile, inflation and credit. Springer Finance
- Caligaris O., Giribone P. G. (2018). Modeling seasonality in inflation indexed swap through machine learning techniques: analysis and comparison between traditional methods and neural networks, Risk Management Magazine, vol 13, issue 3, pp. 37-53.
- Company R., Egorova V. N., Jodar L., Soleymani F. (2018). A local radial basis function method for high-dimensional American option pricing problems, Mathematical Modelling and Analysis, vol 23, N 1, pp. 117-138
- De Prado, M. L. (2018). Advances in Financial Machine Learning, Wiley.
- de Simon-Martin M., Bracco S., Rosales-Asensio E., Piazza G., Delfino F., Giribone P. G. (2020). Electricity Spot Prices Forecasting for MIBEL by using Deep Learning: a comparison between NAR, NARX and LSTM networks, 20th International Conference on Environment and Electrical Engineering (IEEE - EEEIC) 2020 Proceedings, 11th June 2020, Madrid
- Giribone, P. G. (2020). Seasonality Modeling through LSTM Network in Inflation-Indexed Swaps, Data Analytics 2020: The Ninth International Conference on Data Analytics - Special Session: FinTech Risk Management, 25th October 2020, Nice, France
- Heaton J. B., Polson N., Witte J. H. (2017). Deep Learning for Finance: Deep Portfolios, Applied Stochastic Models in Business and Industry, Vol. 33, N. 1
- Hochreiter S., Schmidhuber J. (1997). Long Short-Term Memory. Neural Computing, Vol. 9, No. 8, pp. 1735-1780
- Jarrow R., Yildirim Y. (2003). Pricing Treasury Inflation protected securities and related derivatives using an HJM model, Journal of Financial and Quantitative Analysis, vol. 38, N. 2, pp. 337-358
- Kazziha S. (1999). Interest rate models, inflation-based derivatives, trigger notes and cross-currency swaptions, Ph.D. Thesis, Imperial Collage of Science, Technology and Medicine, London
- Kim P. (2017). “Matlab Deep Learning with Machine Learning Neural Networks and Artificial Intelligence”, Apress
- Lelong J., Lapeyre B. (2020). “Longstaff Schwartz algorithm and Neural Network regression”, Advances in Financial Mathematics, Paris
- Mammadi S. (2017). “Financial time series prediction using artificial neural network based on Levenberg-Marquardt algorithm”, Procedia Computer Science, vol. 120, pp. 602-607

- Mercurio F. (2005). “Pricing inflation-indexed derivatives”, Quantitative Finance, vol 5, no. 3 pp. 289-302
- Pendyala V. (2018). “Veracity of Big Data: Machine Learning and other approaches to verifying truthfulness”, Apress
- Tsay R. S. (2010). “Analysis of Financial Time series” Third Edition, Wiley
- Yanui C., Kaijian H., Geoffrey K. F., “Forecasting crude oil prices: a deep learning based model”, Procedia Computer Science, vol. 122, pp. 300–307