УДК 004.4, 004.6, 004.942, 004.021: 519.6, 336.71 https://doi.org/10.17721/1812-5409.2023/2.38

D.Krukovets, PhD student

Updated DTW+K-Means approach with LSTM and ARIMA-type models for Core Inflation forecasting

Taras Shevchenko National University of Kyiv,
4D, Academician Glushkov ave.,
Kyiv, Ukraine, 03022
e-mail: dkrukovets@kse.org.ua

Д.Ю.Круковець, аспірант

Оновлений DTW+K-Means підхід з LSTM та ARIMA моделями для прогнозування базової інфляції

Київський національний університет імені Тараса Шевченка, Україна, 03022, м.Київ,
просп. Академіка Глушкова, 4Д, 03022
e-mail: dkrukovets@kse.org.ua

The paper is dedicated to evaluating performance in forecasting tasks of the novel routine that includes adapted DTW + K-Means for aggregating series with similar dynamics. The algorithm was developed throughout the series of papers. Novel parts are designed in a way to work with periodic series, like in the investigated monthly data case. It is used over hundreds of Consumer Price Index components to find similar dynamics and aggregate them by the similarity of their dynamics. Then aggregated series are given as input to the ARIMA, SARIMA, and LSTM models, to forecast the total Core Consumer Price Index. The choice is based on the necessity to capture possible non-linear relationships between series.

The dataset is quite rich and contains hundreds of Consumer Price Index components, which is a level of prices for different goods. Data suffers from multiple issues, including seasonality, so controlling them either with satellite models such as X-12 or with the architecture of the forecasting model is sufficient. The research results are important for different groups of agents. Private businesses seek to plan their pricing while government structures want to employ their administrative measures in a proactive data-driven manner.

The result shows that the SARIMA currently outperforms other models. An LSTM model combined with DTW + K-Means method shows worse results yet it was able to catch non-linearities, unlike more traditional models. Further investigation of LSTM + DTW/K-Means performance and fitting is necessary.

Keywords: Dynamic Time Warping, Clustering, ARIMA, Recurrent Neural Networks, LSTM, Forecasting, Inflation

Стаття присвячена оцінці ефективності у проблемах з прогнозування в нової процедури, яка включає адаптований DTW + K-Means підхід для агрегування рядів із подібною динамікою. Алгоритм розроблявся протягом декількох статей. Його новизна полягає в дизайні, що працює з періодичними рядами, місячними в нашому випадку. Ми використовуємо його для агрегації сотень основних компонентів базової інфляції за подібністю їхньої динаміки. Надалі агреговані ряди використовуються в традиційних моделях ARIMA та SARIMA, а також в доволі нестандартній для економетрики, моделі LSTM, для прогнозування загальної базової інфляції. Такий вибір базується на необхідності вловлювати нелінійні відносини між рядами.

Датасет відносно багатий, вміщає сотні компонент Базової Інфляції, себто рівня цін на різні товари. Дані мають низку нюансів, зокрема сезонність, тому контроль в цьому необхідний або додатковими моделями як X-12, або за допомогою, власне, структури прогнозної моделі. Важливість цієї вправи велика з низки причин, залежно від агента. Приватні бізнеси жадають спланувати власне ціноутворення, а уряд намагається впроваджувати політику в стилі, керованому даними, себто проактивно.

Результат вказує, що наразі SARIMA перевершує інші моделі. LSTM в поєднанні з процедурою DTW + K-Means дає гірші результати, проте має здатність вловлювати нелінійності на відміну від традиційних моделей. Подальше дослідження LSTM+DTW/K-Means є необхідним.

Ключові слова: Dynamic Time Warping, Кластеризація, ARIMA, Рекурентні Нейронні Мережі, LSTM, Прогнозування, Інфляція

1. Introduction

CPI forecasting is a widely studied field in econometrics. However, it is continuously expanding with the appearance of new models and approaches every other year. Some of them are based on novel thoughts in the understanding of how economics works. While others are an outcome of the current vector of world development. And this vector is to make as much use of the information around as possible. Enhancing abilities to collect, store and process the data has created an urgent need to develop more and more models that are able to handle this abundance of data and squeeze all the information from it. Some models are quite adaptive, thus economists and computer scientists became able to use them in fields that are not "Big Data". One of those fields will be presented in the paper from a technical rather than economical point of view.

We'll develop a set of forecasting techniques that will be used over the Consumer Prices dataset which includes a lot of components. Some techniques are rather traditional ones, while other are purely from "Data Science" and was fitted to work with large datasets, yet we'll adapt them for our case. Word is about Dynamic Time Wrapping and K-Means to find similarities and aggregate the abovementioned components.

The study of those methods goes through a series of papers over different datasets: artificially created signals in paper [1], deposit rates from Ukrainian banks in paper [2], and nominal wages from regions in paper [3]. The adaptation of DTW and, by the way, one of the key focuses in this paper, lies in altering the algorithm behaviour in the case of a high number of correspondent points to the single point from other series, so as correspondence with a lag over a year. Both of these issues are crucial for high-quality analysis of our series where we'd like to investigate similarity between series with less than a year lag and we'd like to avoid problems that come from long periods of stable inflation for a single component due to natural causes. It's a continuation of the long process of algorithm development.

Then combined components will be used in either the ARIMA/SARIMA routine or RNN/LSTM to make an actual forecast of the aggregated core inflation. In the first case scenario, it will be a set of forecasts that is combined with some weights into the total forecast. Such an approach was used in the paper [4] and it can be called a studied one. In the case of RNN/LSTM, it makes the process of aggregation by itself inside the algorithm, which is one of the powerful parts of the RNN architecture for our purposes. RNN is more exotic in the field, there is a scarcity of papers in the domain of economics that use it and they're quite modern, like papers [5] and [6]. Also, RNN is able to catch non-linearities which are quite common in economics overall which is also studied in the paper [5] in more detail. The usage of Data Science techniques in the economic field, central banking, in particular, is discussed in the meta-analysis paper [7].

The question about seasonality appears here and is dealt with using the X-12 technique, described in the paper [8], for the ARIMA model, naturally for SARIMA and RNN models. SARIMA model is defined in a way to capture lag that corresponds to the frequency of the data (12 for monthly) and takes into account this lag either with AR or MA component, or both. RNN has a long-term memory in the formula and a number of layers that allow to extraction of information from many quarters before the actual moment.

But why is this exercise so important in real life? Because inflation is a crucial parameter for private agents and government, Central Banks, in particular, maintain their monetary policy. Private agents want to understand the price level in the future and plan their current pricing policy, enhancing or decreasing production and have to find an answer to other questions. From the CB point of view, in many countries, including Ukraine, the monetary framework is called "Inflation targeting" which has, among several other goals, an obligation to maintain price stability and have inflation at its target level, which is given in statement [9]. Thus, deviations in inflation make Central Banks use their instruments, which work in the medium-to-long run in order to put back the inflation to its target value [10].

2. Data

The dataset resembles the long-standing tradition in my series of research to originate from the economics domain like in [2], [3], [4]. In this research, the Core Consumer Price Index (Core CPI) will be investigated and corresponding forecasting models will be built. But let us dive deeper into the data with the explanation of the Core CPI.

Consumer Price Index (CPI) is a measure of all tradable goods in the economy, starting from simple vegetables and fruits, going with clothes and different electronics, and finishing with services, including electricity bills or university prices. The variety of those goods is not that far from infinite, thus the State Statistical Service of Ukraine, namely Ukrstat, collects those prices for "standard" goods and also their part in the basket of an average Ukrainian. This methodology is written in more detail in the [11]. It allows for observing the UAH cost of living for an average Ukrainian and gives the ability to calculate many administrative entities such as minimal wage, the minimum cost of living (living wage), yearly pension adjustment and others.

Core CPI is a sub-part of the total CPI that is dedicated to components, that are less dependent on administrative decisions (such as the cost of electricity), weather (raw food) or world fluctuations (petrol price, that is dependent on oil prices). It has more market-based dynamics than other components which supports an ability to make data-driven forecasting models that are more resilient to exogenous shocks than other components. Core CPI has two major divisions: four sub-categories that include Processed foods (like sausages), Clothes, Services and Others; so as more than 300 components that have a weight in an average Ukrainian consumer basket of more than 0.1%. The amount of components is re-calculated every five years. Both divisions will be actively used in upcoming models.

The data will be taken on a monthly basis from 2012m1 for the division with 4 components and from 2007 for the division with many components. This comes from the availability of the data and completeness of methodologies to combine those components into 4 major subcomponents. The data is available on a monthly basis.

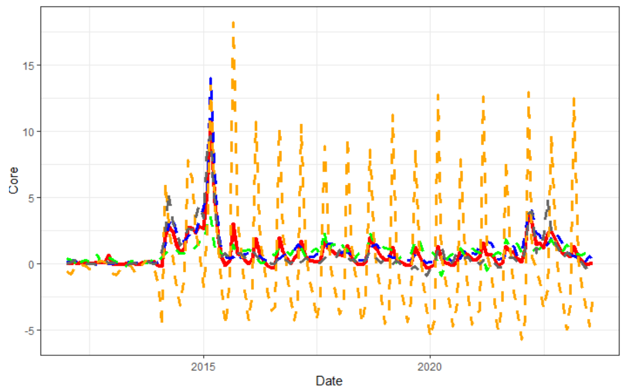
One of the key laws for obtaining a good forecasting model is the stationarity of the series. To take one step in the direction of ensuring stationarity of all components, at least weak stationarity, we can take a percentage change of those Core CPI levels and obtain a Core Inflation. While it's merely possible to ensure stationarity for all components, the nature of the series becomes stationary and tests for stationarity may fail due to shocks in those series rather than due to the natural non-stationarity. Overall, we're obtaining 140 data points that start from 2012m1 to 2023m8 for four major components and 200 data points that start from 2007m1 to 2023m8 for all 300+ components.

The first question that may arise in the mind of a reader is if we forecast those disaggregated components, how we could aggregate those components back to the overall forecast of the Core CPI. The answer is simple yet intricate. As long as we can use weights proposed by Ukrstat, they may slightly vary throughout the time and also be hardly obtainable for approaches, that are based on aggregating hundreds of components. Thus, a more plausible way to aggregate those series will be given in a Model section as a supplementary model.

Another issue that may be present in the data (and, to be honest, is actually present) is seasonality.

Inflation and its components have strong seasonal patterns. Some are justified by design: prices for universities are usually recalculated right before the new study year begins, while clothes have new collections for summer and winter in spring and autumn respectively. Electronics have regular sales in December, right before New Year's Eve and hotels increase their prices for a summer season and/or winter. Partially it's described in the [12]. We can have a closer look at the Core Inflation and its components in Figure 1. It clearly states a seasonality for most of the components, especially for clothes. Also, there was a major shock in 2015 right after the first invasion from rusnya into Ukraine and the start of the rusnya-Ukraine war. But overall inflation seems to be quite stable over time.

Figure 1. Core Inflation (red) and its 4 sub-categories: Processed Food (blue), Clothes (orange), Services (green), Other (grey)



Another look at the same problem may be done using the AutoCorrelation Function and Partial AutoCorrelation Function (ACF/PACF). This approach uses a correlation between the series and its lagged version in order to describe the relationship of the time series with itself. Large autocorrelation at some period means that they're dependent. In terms of seasonality large coefficient in the corresponding frequency (12 in our case) means the presence of this stationarity. The formula is given below:

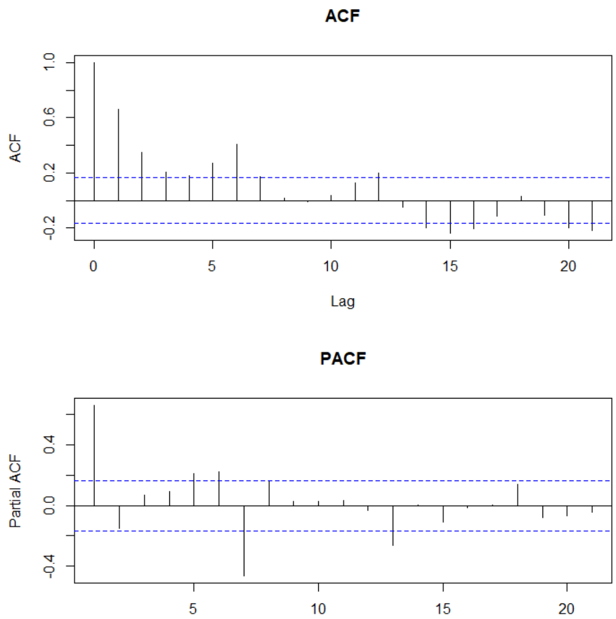
$$\rho_k = \frac{\sum_{t=k+1}^{n}(y_t - \bar{y})(y_{t-k} - \bar{y})}{\sum_{t=1}^{n}(y_t - \bar{y})^2}$$

Where y_t is a value in period t, $\bar{y}$ is an average.

PACF has a similar concept yet it's cleaned from the endogeneity with other ACF coefficients.

It shows in Figure 2 that Core Inflation has a significant (yet not that large) seasonality in ACF in both lag 6 and 12, while PACF shows seasonality in lag 6 only. Half-annual seasonality is an uncommon pattern, yet it may be partially driven by the biannual nature of clothes price dynamics.

Figure 2. ACF and PACF for Core Inflation



To deal with this seasonality, we'll employ either tools for seasonal adjustment such as the classic X-12 procedure or utilize models that count seasonality by design (SARIMA). Yet, the main goal is to forecast a seasonally unadjusted Core CPI, thus the seasonal factor from the previous year before forecasting will be returned after all the necessary modelling and forecasting routine.

The last but not least problem is a part of the dataset with 300+ components. As it was mentioned before, some of them have actually been added by Ukrstat way later than 2007 which leads to an abundance of NA. There are multiple solutions to this problem, but the simplest and the chosen one is to simply delete all columns with NA and rescale weights which dropped down the total amount of components from 335 to 270. This dataset was extensively described in the paper [4],

To summarize, the dataset consists of monthly Core CPI components from either 2007m1 or 2012m1 until 2023m8. There are issues with stationarity that are partially solved by using Core Inflation (the first difference of CPI) and seasonality, which require either seasonal adjustment tools or specific models (SARIMA). The next part will take a deeper dive into actual models that are built for forecasting purposes.

3. Model

Now we can proceed to the set of models, which represent classic models used in Time Series analysis reinforced with more sophisticated Recurrent Neural Network, namely LSTM. All of them are used over the set of Core Inflation components, either a traditional set with processed food, clothes, services and others or more sophisticated, obtained from 270 components on a lower level of disaggregation and combined with a novel DTW+K-Means approach investigated throughout series of papers. For both approaches if used with ARIMA-type models, we've got to aggregate them back to the Core Inflation which can be done using a supplementary OLS model, that calculates necessary weights on the history and can be used in the forecast.

3.1. ARIMA and SARIMA

Let's start with a standard Autoregressive Integrated Moving Average (ARIMA), a widely utilized time series forecasting method in a majority of fields that are related to time series analysis. It's a go-to model for most econometric and time series analysis research so for Data Science / Big Data that includes working with signals as a reference model, see [4], [13], [14], [15], [16]. It's known for its capability to capture temporal dependencies and patterns within sequential data. ARIMA amalgamates three key components: autoregressive (AR), differencing (I), and moving average (MA) processes. The AR component elucidates the relationship between a data point and its lagged observations, with the order denoted as 'p' representing the number of lagged values considered in the prediction. Differencing, encompassing an order denoted as 'd', is employed to attain stationarity by subtracting each data point from its preceding value, effectively mitigating trends or changing means. The MA component characterizes the relationship between the current data point and past white noise or error terms, with the order indicated as 'q', representing the number of lagged error terms utilized in prediction. The general formula is given below:

$$(1 - \phi_1 L - \phi_2 L^2 - \ldots - \phi_p L^p)(1 - L)^d X_t = (1 + \theta_1 L + \theta_2 L^2 + \ldots + \theta_m L^{\varphi}) Z_t$$

Where X_t is a series in a period t, Z_t is an error in a period t, L is a backward shift operator, ϕ_p and θ_m are estimated coefficients.

As it was stated in the data section, series may impose some seasonality issues and there are two possible solutions, both of which will be used in this research. The first one is a seasonal adjustment of the series with an X-12 approach. It's a traditional approach for dealing with seasonality issues. The underlying principle of the X-12 algorithm involves a systematic process of decomposition and adjustment to discern seasonal patterns and other irregularities in time series data. This algorithm sequentially follows a series of steps to achieve this goal. Firstly, the algorithm decomposes the time series into distinct components, namely the seasonal, trend, cyclical, and irregular components. Then it identifies the seasonal component by analyzing recurring patterns within the data at regular intervals, within a year. This step involves the detection of periodic variations to isolate the seasonal effects. Once the seasonal component is identified, the algorithm proceeds to perform seasonal adjustment. This entails removing the seasonal effects from the time series, typically using methods such as the X-11 method, which employs statistical techniques like moving averages to smooth out the seasonal fluctuations. It is fully described with way more details in the paper [8],

The second approach to work with seasonality issues is to employ the Seasonal ARIMA (SARIMA) model. This model works like a regular ARIMA, yet it has also seasonal p, d, and q terms which are based on a series frequency (monthly in our case, thus the seasonal lag will be 12) and add to the standard ARIMA formula a seasonal part (12-th lag). It is a great approach for a data-driven model to count for seasonality. The general formula is given below:

$$(1 - \phi^1 L - \phi^2 L^2 - \dots - \phi_p L^{\mathrm{p}}) * (1 - L) * (1 - \Phi^1 L^s - \Phi^2 L^{2s} - \cdots - \Phi_p L^{ps}) * (1 - L^s)^{\mathrm{D}} X_t = (1 + \theta^1 L + \theta^2 L^2 + \dots + \theta_m L^{\varphi}) * (1 + \Theta^1 L^s + \Theta^2 L^{2s} + \dots + \Theta_m L^{qs}) * Z_t$$

Where X_t is a series in a period t, Z_t is an error in a period t, L is a backward shift operator, ϕ_p and θ_m are estimated coefficients, s is a seasonal frequency which defines the lag that should be taken (12 for monthly series).

This is less widely spread as a go-to model than a standard ARIMA. Nonetheless, it shows how useful the approach is where the seasonality is an issue like in the paper [17].

In both cases, the specifications, namely p, d, q, P, D, Q are chosen using an auto.arima() procedure. The procedure is based on the Akaike Information Criteria (AIC) and checks models with all combinations of p, d, q, P, D, Q in a grid-search manner. It estimates AIC and chooses the model with the smallest one. The AIC, unlike the SSE (Sum of Squared Errors), punishes for too many lags taken in the model so the ones that have a lot of coefficients without much improvement in the likelihood are avoided. The formula for AIC is as follows:

$$AIC = 2k - 2\ln(L)$$

Where k is the number of coefficients, L is the maximum likelihood of the model with the corresponding set of coefficients.

The approach to estimate coefficients is better described in the paper [18].

3.2. RNN/LSTM

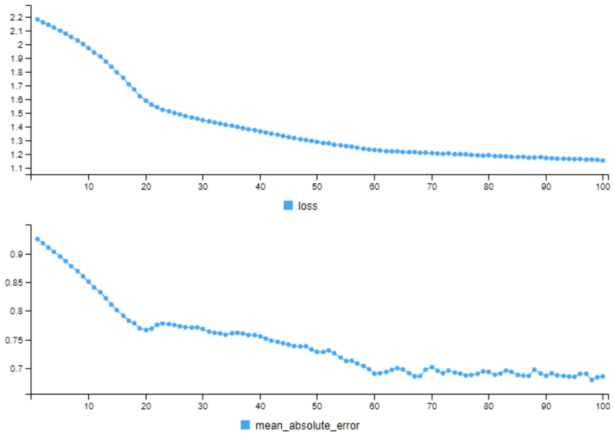
Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks are recurrent neural network architectures primarily used for processing time series data. In the latest econometric research, we can observe rising interest in Data Science methods, including RNN and other neural networks like in [5], [6].

RNN and LSTM are slightly different in functionality. In a standard RNN, each neuron receives input from the preceding time step and generates an output for use in the subsequent time step. However, RNNs struggle to capture long-term dependencies effectively due to the "vanishing gradient" issue, where gradients become quite small. This is strongly unacceptable in the case of economic time series as long as they're quite dependent on longer-term interactions, especially with present seasonality, and require slightly different approaches.

Conversely, LSTMs were developed to overcome the vanishing gradient problem inherent in RNNs. LSTMs incorporate a more intricate architecture, including a cell state and three gated mechanisms (input, forget, output gates). These gates control the information flow, allowing LSTMs to sustain and update information over more extended sequences, making them proficient in capturing long-term dependencies. This is crucial for a good performance in the current research thus it is a go-to model for us in the pipeline as an alternative to traditional ARIMA models. For more information about the difference between RNN and LSTM architectures, one can refer to [19].

Classic LSTM training routine may be seen in Figure 3 where the model runs through 100 epochs of training. After multiple experiments, we decided to implement a simple LSTM with one LSTM layer, one flattening layer, and two dense layers, one of which is an output layer with a single output value, while other layers have a number of outputs equal to the amount of input series. The activation function is a simple RELU.

Figure 3. The training process of the LSTM model minimizes loss and MAE.



The model is trained on the dataset where X values are taken with the corresponding lag (depending on the exercise, whether we're talking about forecasting 1-month ahead or 6-months ahead). Y values are taken without the lag in order to make the model able to, actually, forecast with the available data. Then, in the test exercise, the last row of available data is taken to produce a forecast on 1 or 6 months ahead depending on the model.

3.3. DTW + K-Means

A crucial part of the research, one of the major interest points, is the novel algorithm that uses an adapted Dynamic Time Warping algorithm with a K-Means approach in order to investigate the similarity between CPI components and divide them into groups according to the abovementioned similarity. This allows to make groups not by economic logic, but by the data-driven logic which may observe patterns that are unable to be seen without a detailed investigation of those components.

In its initial form, the Dynamic Time Warping (DTW) algorithm creates a matrix that showcases the distances between each data point in one sequence with every data point in another sequence. The bottom-left cell corresponds to the distance between the first data points, while the upper-right cell signifies the distance between the final data points. Subsequently, a path is laid out, connecting these cells in a manner that minimizes the sum of distances. This approach not only helps prevent the overlap of correspondences but also links all data points. It accommodates situations where there might be time shifts between corresponding points (lagged reactions), instances of stretching (where a single point corresponds to multiple points), and contraction (which is the inverse of stretching). The algorithm and its implementation in R is fantastically described in [20].

One of the most notable algorithmic improvements is the introduction of FastDTW, as presented in [21]. FastDTW was developed to expedite the computation process. The original algorithm has a complexity of O(n), resulting in a quadratic increase in processing time as the number of observations grows. FastDTW's key insight is that it is unnecessary to compute the entire matrix; only a partial calculation is required. This revelation stems from the observation that the primary path tends to reside in the central region. Consequently, a few "masks" are utilized to "shade" the area that is less likely to contain a part of the path, and therefore, should not be computed. While FastDTW incorporates several other enhancements, the aforementioned one stands out as pivotal due to its illumination of the limitations of the distance matrix. It is worth noting that, for this study's purposes, the widespread adoption of FastDTW is not essential, as the focus is on relatively small datasets that do not exceed a thousand observations.

However, ideas built-in in the FastDTW gave birth to another modification specific to this research. It is the adjustment that disallows long stretches, where numerous similar values correspond to a single value, in line with the classic algorithm. This change was implemented to address an issue that arises with certain data components exhibiting unusual dynamics. In such cases, a single peak could correspond to almost every data point, while a relatively stable period of 11 data points could correspond to only a few points in another series. This phenomenon is particularly observed in datasets related to university tuition prices and funeral service costs, which exhibit strong seasonality patterns in particular periods once or twice a year.

To work out this issue we're adding to the Cost Matrix, which is necessary to build a correspondence between series, the restriction to have which doesn't allow either i or j to be the same more than 10 times in a row. This is done with a dynamic mask at each step in a similar manner as it is done in the FastDTW approach. The formula for the Cost Matrix is below:

$$C(i,j) = d(x_i, y_j) + min\{C(i-1,j), C(i,j-1), C(i-1,j-1)\}$$

After computing distances using the methods described above, these distances can be organized into corresponding matrices. Each cell in row p and column q represents the distance between series p and q. This matrix demonstrates symmetry, as the distance between p and q is equivalent to the distance between q and p.

Having obtained a two-dimensional plane with a set of points during the preceding stage, we can now employ a clustering technique to group similar series into a single cluster. The chosen algorithm for this purpose is the simple K-Means clustering. This choice is motivated by its simplicity and interpretability, which aid in hypothesis checking, align with basic visual analysis results, and more. Also, other papers in the series, namely [1], [2], [3], have shown that the performance of K-Means consistently surpasses all the other algorithms, namely DBSCAN, Hierarchical Clustering and their sub-algorithms.

3.4. OLS for weights

The last part of the section corresponds to the aggregation problem in the case of ARIMA forecasting. This way we'd obtain a set of forecasts for components (the number depends on the choice of whether we're forecasting with DTW+K-Means or using predefined components). We've got to aggregate them into the total forecasts, which may be computationally cumbersome for the case with DTW as long as we've deleted 65 components due to their incompleteness (see Data section for the discussion).

To do things simply yet effectively, there is a possibility to employ simple OLS on the history to find what are the correct weights to aggregate components. Obviously, as the economy changes, the weight set changes too because people buy more new things and less of the old. Also, when countries get richer, people spend less of their income (in per cent) on food and basic needs and more on services. Yet these weights change slowly and not that strongly, so we may assume that they're stable and use OLS-based estimates.

The formula is as follows:

$$\pi_t^{core} = \beta_1 * \pi_t^{component^1} + \beta_2 * \pi_t^{component^2} + \cdots + \beta_p * \pi_t^{component^p}$$

Where π_t is an inflation, core or component correspondingly; β_p is an estimated coefficient for component p.

Found coefficients are then used for the forecasted components' inflation to aggregate them back into the forecast of the Core Inflation.

3.5. Summary

To finish the section, a short reminder of the pipeline. We're running 6 models in total, divided into two groups. The first group is based on 4 predefined Core Inflation components, the second is based on components found by the DTW+K-Means procedure. Inside each group, there is the ARIMA model with X-12 seasonal adjustment, the SARIMA model, and LSTM. In the case of the former two, after producing the actual forecasts for components, we have to aggregate them back into the total Core Inflation. To do this, we employ OLS that can find weights for each component and then those weights are multiplied by each component inflation and summed up into the whole forecast.

4. Results

4.1. Results of models

Now it's time to utilize all the data and models to produce some high-quality forecasts of Core Inflation. But first, we've got to ensure the quality of those models. To do so, we'll estimate models on multiple databases in a shifting window fashion. It means that we'll use the data for up to some period as a training dataset. Then we'll produce a pseudo-out-of-sample forecast for 1 or 6 months ahead. After this, we'll expand the dataset by a single time period and repeat the exercise. This process will start from 2021m1 to 2023m7 for 1m ahead forecasts and 2023m2 for 6m ahead forecasts. All the pseudo-forecasts are combined in a vector and compared with actual values using Root Mean Squared Error. RMSE is a standard way to ensure the forecast quality and is calculated with a formula:

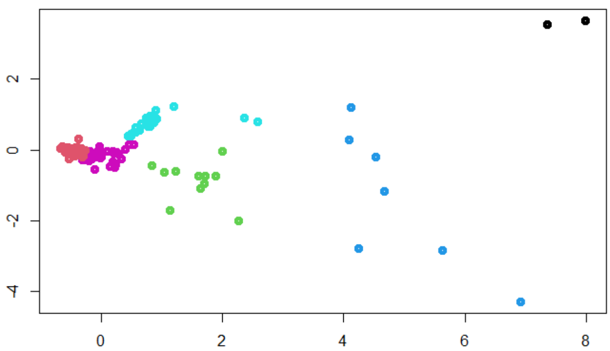
$$RMSE = \sqrt{\frac{\sum(x_t - \hat{x_t})^2}{n}}$$

Where x_t is an actual value and $\hat{x_t}$ is the predicted value, and n is the number of points.

Such an approach allows us to count for strong and small deviations, assuming we want a model that does not fail strongly and is rather prone to medium errors at the time.

But first, we'll go through the result for the DTW+K-Means routine for Core CPI components that are used in half of the shown models. Mapping the dynamics of the series on a 2D plot allows us to see which series are close to which and put them into groups. In Figure 4, we can observe that most series are inside three large clusters and overall dynamics in those series are relatively close, while there are several other outlier groups.

Figure 4. Core Inflation components are represented as points in the 2D plane and grouped according to the K-Means algorithm



The following stage is to actually fit a set of models and use them for forecasting purposes as it was described in section 3. Overall, we'll create 12 comparisons of 3 models with 2 inputs each (no-DTW and DTW) and 2 forecasting horizons, 1 month and 6 months ahead. The results are in Table 1 below. A general robustness check in the forecasting modelling is that RMSE for further periods is worse (larger) than RMSE for closer periods. A check mark is here. LSTM generally shows poorer performance than ARIMA models which may come from not enough work fitting the parameters, because opportunities in LSTM architecture are wide while when we're talking about ARIMA/SARIMA models they're nearly perfect already. Also, the NoDTW approach in this case shows better performance, except in the case of the LSTM model. There are multiple explanations for this phenomenon, including that we're in a war now and this is a huge shock which may be misinterpreted with simpler ARIMA models yet caught by the more complex LSTM when we're talking about DTW preparation. While a good analyst will choose the SARIMA model without DTW which outperforms all other models in both 1-month ahead and 6-month ahead problems, a good modelling specialist will consider further experiments with LSTM + DTW design to find the specification that might beat other competitors. Also, the comparison with standard deviation is a great way to understand how good models are because if the deviation is less than a standard deviation, we are likely to be in the region with the most probability for a new observation to be there.

Table 1. RMSE comparison of models.

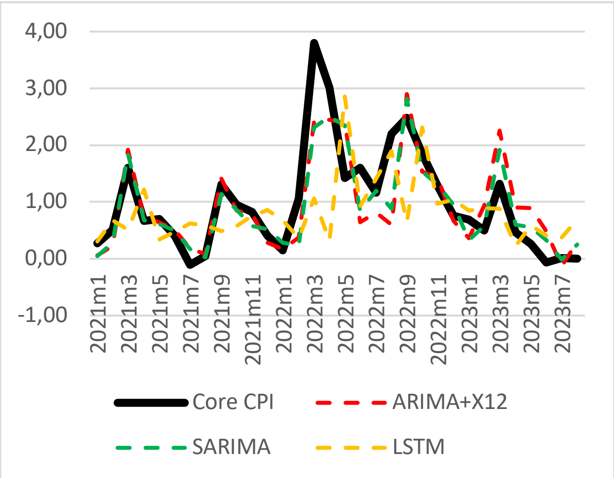
RMSE	ARIMA + X12	SARIMA	LSTM
NoDTW, 1m	0.561	0.493	0.914
DTW, 1m	0.689	0.649	0.791
NoDTW, 6m	0.912	0.862	1.219
DTW, 6m	0.921	0.956	1.135
Standard deviation		0.921	

Continuing an investigation of the result, let's build a plot with pseudo-forecasts and actual data. We can see how well fluctuations are captured by 1m ahead models in Figures 5-8. An intricate thing is happening with the difference between DTW and non-DTW in terms of ARIMA models, which capture worse the peaks in the second case while working quite well in relatively stable periods. It may be a signal that the weight of the series that captures most reactions on war shocks is too small, as long as this weight was estimated in the period without a full-scale invasion. As for the performance of the LSTM model, it performs well not that regularly, failing in many scenarios based on seasonality, which means that improving the LSTM model with seasonality-capturing algorithms or extensions is crucial for its performance.

When we're talking about the second set of graphs, Figures, for 6 months ahead forecasts, in the case of no DTW we can observe strong reliance on seasonal patterns instead of our forecasting ability for all models including LSTM while in the latter case, with DTW, models don't rely much on seasonality at all. It may be in favour of the latter models when we're talking about times when actual shocks have to be predicted instead of a standard seasonality.

Anyway, throughout all the models none of them has been able to predict a strong shock with a full-invasion start which means that components barely have that signal about the worsening of the economic situation. It may lead to two thoughts for improving the set of models. Add exogenous variables that may be a leading indicator for upcoming crises, for example, change in the risk premia or estimate model quality on periods such as 2019m1-2021m12. Both of them have their pros and cons yet this may amplify the quality of understanding how the model performs or the quality of the model itself.

Figure 5. Core CPI and its 1-month ahead forecast according to models without DTW.



At the end of the section, we'd like to present a simple forecast for 1 month ahead, 2023m9 to be more precise, of the Core Inflation by 3 models and DTW. In Figure 9, we can clearly see the superiority of more standard ARIMA approaches in capturing the dynamics yet it may come from a better ability to capture seasonality than in the LSTM model, which is high for the Core Inflation each September. This is one more piece in a story that LSTM should be reinforced with extensions to deal with the seasonality issues.

Figure 6. Core CPI and its 1-month ahead forecast according to models with DTW.

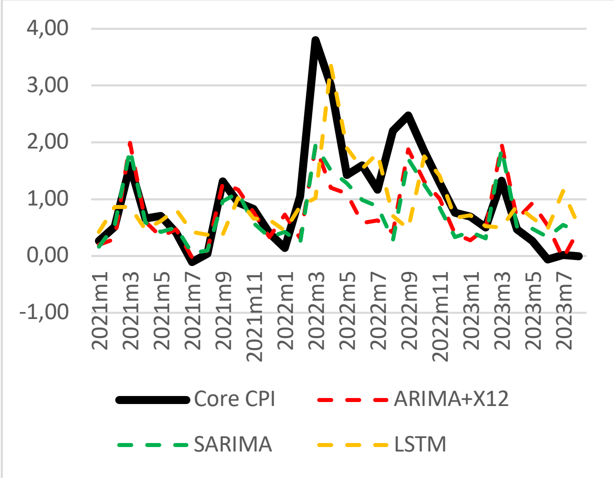


Figure 7. Core CPI and its 6-month ahead forecast according to models without DTW.

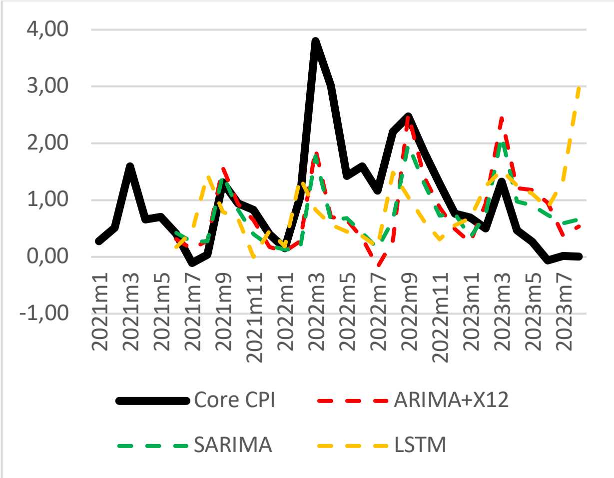


Figure 8. Core CPI and its 6-month ahead forecast according to models with DTW.

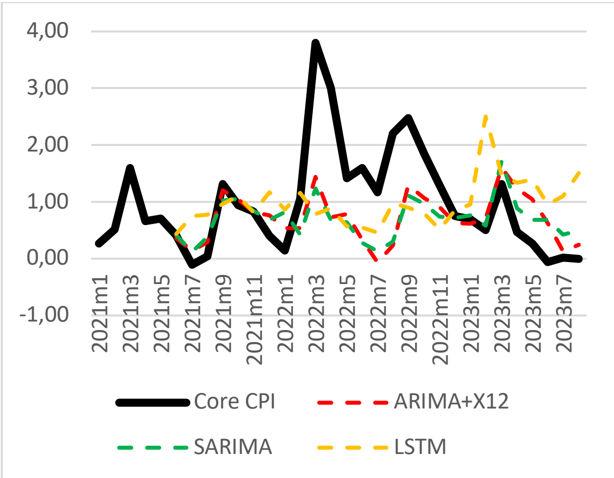
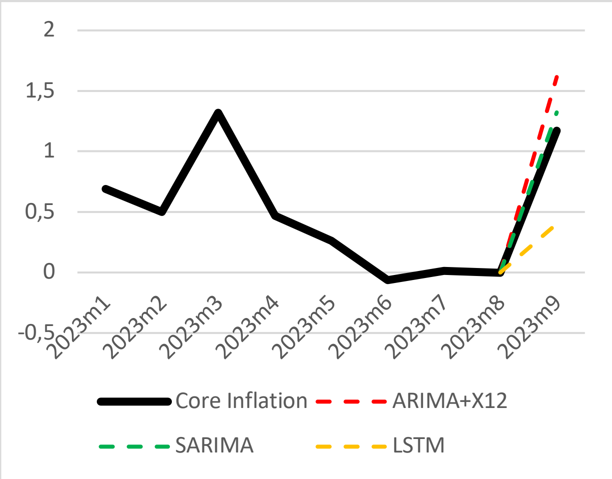


Figure 9. One month ahead forecast (2023m9) for all three models with DTW procedure.



4.2. Discussion

A majority of estimated models show a decent performance in comparison with standard deviation. In figures from 4.1. they show an ability to capture patterns from the previous data yet adapt them according to the upcoming situation. This comes from an ability to extract signals from the aggregated data.

Those figures in 4.1. and the corresponding discussion shows that seasonality is an issue that requires special attention in the upcoming research and may lead to the ability of LSTM architecture (along with additional fitting) to outperform SARIMA models.

A traditional way to investigate the model performance further is to compare them with state-of-the-art models in the domain. However, there are multiple issues with a comparison between these estimated models and the ones from other literature. Firstly, the estimation is made for Ukrainian data only which has a lot of specifics compared with other countries (ongoing war is not the only problem) and models that are well for other countries may perform poorly in Ukraine. Secondly, the estimation window captures 2021-2023 years which is quite modern and the Core CPI itself has a higher volatility here than, namely, in 2017-2019 making results from older papers incomparable. Thirdly, the methodology, rolling-window RMSE, is crucial for a good estimation yet not used in a majority of papers for Ukraine making it impossible to compare model quality.

One option for a better comparison is to estimate these models and benchmark models (RW, AR1) both on the Ukrainian dataset and other countries' datasets and make an overwhelming comparison between countries and models, including a direct comparison of model quality from other papers, which actually exists (even in [5], see table 4).

To summarize, we've built 6 models and estimated their quality in forecasting 1-month ahead and 6-month ahead using pseudo-out-of-sample forecasts in 2021m1-2023m8, corresponding graphs and the RMSE table. LSTM shows the worst performance yet it may still be a promising model that carries a wide field of extensions for architecture, dealing with seasonality and other. At the moment, if any need for Core inflation forecasting occurs, the recommendation will be to use the SARIMA model without DTW.

5. Conclusion

The paper goes through a data-driven set of models to forecast Core Inflation. At the moment, the most promising one is a SARIMA based on four predefined components of Core Inflation: Processed Food, Clothes, Services and Others. It gives relatively great RMSE over the span of 2021m1-2023m8, where it was tested in forecasting 1 month ahead and 6 months ahead. Such a result is unfavourable for a novel approach with adapted DTW and K-Means, yet there are a number of signs that this approach paired with better fitted LSTM model may give even better results due to the ability to catch non-linear dependencies unlike the standard ARIMA models, reinforced with OLS-based weights for aggregating.

As for further development, we'll try slightly other specifications of adapted DTW+K-Means routine, so as other architectures for LSTM model with different layers. Another thing is to use seasonally-adjusted series in the LSTM input and not rely on it in terms of the ability to catch the seasonal component.

References

1. KRUKOVETS, D. (2020): *Analysis of similarity between artificially simulated time series with Dynamic Time Warping.* "Proceedings of Workshop on Intelligent Information Systems WIIS2020", pp.97-108.
2. KRUKOVETS, D. (2022): *Multi-stage approach with DTW and clustering for forecasting of average deposit rate in Ukraine.* "Bulletin of Taras Shevchenko National University of Kyiv Series Physics and Mathematics", pp.55-65
3. KRUKOVETS, D. (2021): *Dynamic Time Warping for uncovering dissimilarity of regional wages in Ukraine.* "Proceedings MFOI-2020", pp.168-185
4. KRUKOVETS, D., VERCHENKO, O. (2019): S*hort-run forecasting of core inflation in ukraine: a combined arma approach.* "Visnyk of the National Bank of Ukraine 248", pp.1-10-
5. ALMOSOVA, A., ANDRESEN, N. (2019): *Nonlinear Inflation Forecasting with Recurrent Neural Networks.* "European Central Bank (ECB)", pp. 1-45.
6. LONGO, L., RICCABONI, M., RUNGI, A. (2022): *A neural network ensemble approach for GDP forecasting.* "Journal of Economic Dynamics and Control"
7. KRUKOVETS, D. (2020): *Data science opportunities at central banks: overview.* "Visnyk of the National Bank of Ukraine 249", pp.13-24.
8. EUROPEAN CENTRAL BANK (ECB) (2007): *Guide to Seasonal adjustment with -12-ARIMA, March 2007,* pp. 1-169.
9. THE COUNCIL OF THE NATIONAL BANK OF UKRAINE (2018): *Monetary Policy Strategy Of The National Bank Of Ukraine,* pp. 1-9.
10. TENREYRO, S. (2023): *Macroeconomic stabilisation in a volatile inflation environment.* "ECB forum on Central Banking", pp. 1-37.
11. PROFATSKA, N. (2021): S*tandard quality report state statistical observation "changes in prices (tariffs) for consumer goods (services)" 2.06.01.01.* "State Statistics Service of Ukraine", pp. 1-11.
12. STOCK, J., WATSON, M. (2019): *Trend, Seasonal, and Sectoral Inflation in the Euro Area.* "Princeton University", pp.1-32.

Список використаних джерел

1. Krukovets, D. Analysis of similarity between artificially simulated time series with Dynamic Time Warping. / D. Krukovets // Proceedings of Workshop on Intelligent Information Systems WIIS2020. – 2020. – P. 97-108.
2. Krukovets, D. Multi-stage approach with DTW and clustering for forecasting of average deposit rate in Ukraine. / D. Krukovets // Bulletin of Taras Shevchenko National University of Kyiv Series Physics and Mathematics. – 2020. – P. 55-65
3. Krukovets, D. Dynamic Time Warping for uncovering dissimilarity of regional wages in Ukraine. / D. Krukovets // Proceedings MFOI-2020. – 2021. – P. 168-185
4. Krukovets, D., Verchenko, O. Short-run forecasting of core inflation in ukraine: a combined arma approach. / D. Krukovets // Visnyk of the National Bank of Ukraine 248. – 2019. – P. 1-10
5. Almosova, A., Andresen, N. Nonlinear Inflation Forecasting with Recurrent Neural Networks. / A. Almosova, N. Andresen // European Central Bank (ECB). – 2019. – P. 1-45.
6. Longo, L., Riccaboni, M., Rungi, A. A neural network ensemble approach for GDP forecasting. / L. Longo, M. Riccaboni, A. Rungi // Journal of Economic Dynamics and Control – 2022.
7. Krukovets, D. Data science opportunities at central banks: overview. / D. Krukovets // Visnyk of the National Bank of Ukraine 249. – 2020. – P. 13-24.
8. European Central Bank (ECB) Guide to Seasonal adjustment with -12-ARIMA, March – 2007. – P. 1-169.
9. The Council of The National Bank Of Ukraine Monetary Policy Strategy Of The National Bank Of Ukraine – 2018. – P. 1-9.
10. Tenreyro, S. Macroeconomic stabilisation in a volatile inflation environment. / S. Tenreyro // ECB forum on Central Banking. – 2023. – P. 1-37.
11. Профацька, Н. В. Стандартний звіт з якості державного статистичного спостереження "зміни цін (тарифів) на споживчі товари (послуги)" 2.06.01.01 / Н. В. Профацька // Державна служба статистики України. – 2021. – Ст. 1-11.
12. Stock, J., Watson, M. Trend, Seasonal, and Sectoral Inflation in the Euro Area. / J. Stock, M

Watson // Princeton University. – 2019. – P.1-32.

13. SHAPOVALENKO, N. (2021): *A Suite of Models for CPI Forecasting.* "Visnyk of the National Bank of Ukraine", pp. 252,4-36.
14. SIAMI-NAMINI, S., TAVAKOLI, N., SIAMI-NAMINI, A. (2018): *A Comparison of ARIMA and LSTM in Forecasting Time Series.* "17th IEEE International Conference on Machine Learning and Applications", pp. 1-8.
15. KUMAR, M., THENMOZHI, M. (2014). *Forecasting stock index returns using ARIMA-SVM, ARIMA-ANN, and ARIMA-random forest hybrid models.* "International Journal of Banking Accounting and Finance 5(3):284"
16. FAN, G., ZHANG, L., YU, M., HONG, W., DONG. S. (2022). *Applications of random forest in multivariable response surface for short-term load forecasting.* "International Journal of Electrical Power & Energy Systems, Volume 139"
17. VALIPOUR, M. (2014): *Long-term runoff study using SARIMA and ARIMA models in the United States.* "Meteorol. Appl. 22", pp. 592–598.
18. HYNDMAN, R., KHANDAKAR, Y. (2008): *Automatic Time Series Forecasting: The forecast Package for R.* "Journal of Statistical Software", 27(3), pp. 1-22.
19. SHERSTINSKY, A. (2018): *Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network.* "Cornell University", pp. 1–43.
20. GIORGINO, T. (2009): *Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package.* "Journal of Statistical Software", 31(7), pp. 1–24.
21. SALVADOR, S., CHAN, PH. (2007): *Toward accurate dynamic time warping in linear time and space.* "Intelligent Data Analysis", 11(5), pp. 561-580.

13. Shapovalenko, N. A Suite of Models for CPI Forecasting. / N. Shapovalenko // Visnyk of the National Bank of Ukraine. – 2021. – P. 252,4-36.
14. Siami-Namini, S., Tavakoli, N., Siami-Namini, A. A Comparison of ARIMA and LSTM in Forecasting Time Series. / S. Siami-Namini, N. Tavakoli, A. Siami-Namini // 17th IEEE International Conference on Machine Learning and Applications. – 2018. – P. 1-8.
15. Kumar, M., Thenmozhi, M. Forecasting stock index returns using ARIMA-SVM, ARIMA-ANN, and ARIMA-random forest hybrid models. / M. Kumar, M. Thenmozhi // International Journal of Banking Accounting and Finance 5(3):284. – 2014.
16. Fan, G., Zhang, L., Yu, M., Hong, W., Dong. S. Applications of random forest in multivariable response surface for short-term load forecasting. / G. Fan, L. Zhang, M. Yu, W. Hong, S. Dong // International Journal of Electrical Power & Energy Systems, Volume 139. – 2022.
17. Valipour, M. Long-term runoff study using SARIMA and ARIMA models in the United States. / M. Valipour // Meteorol. Appl. 22 – 2014 – P.592-598
18. Hyndman, R. Automatic Time Series Forecasting: The forecast Package for R. / R. Hyndman, Y. Khandakar // Journal of Statistical Software. – 2008. – Volume 27 – Issue 3 – P. 1-22.
19. Sherstinsky, A. Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network. / A. Sherstinsky // Cornell University. – 2018. – P. 1–43.
20. Giorgino, T. Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package. / T. Giorgino // Journal of Statistical Software. – 2009. – Volume 31 – Issue 7 – P. 1-24.
21. Salvador, S., Chan, Ph. Toward accurate dynamic time warping in linear time and space. / S. Salvador, Ph. Chan // Intelligent Data Analysis. – 2007. – Volume 11 – Issue 5 – P. 561-580.

Надійшла до друку 20.10.2023