Empowering Time-Series Forecasting in Official Statistics through Transformers

Alberico Emanuele[1], Francesco Pugliese[2], Massimo De Cubellis[2], Angela Pappagallo[2]

[1]*Whitehall Reply, Via del Giorgione, 59, Rome, 00147, Italy*
[2]*Italian National Institute of Statistics – Istat, Via Cesare Balbo, 16, Rome, 00184, Italy*

Abstract

Artificial Intelligence (AI) is playing a crucial role in the promotion of innovation in public administrations. Extensive research and studies on the use of AI to support and improve traditional statistical production processes have been carried out at Istat. This paper presents a pioneering approach based on Transformer neural networks for forecasting time series. This experiment confidently applies a neural network from Natural Language Processing to a new context, specifically for predicting time series. The experiment analyzes four indicators of significant socio-economic interest, namely Gross Domestic Product (GDP), unemployment rate, inflation, and consumer confidence rate, using both Transformers and traditional methods and models. This paper provides a comparative analysis between the performance of Transformers and other statistical methods used in the context of time series forecasting, such as Auto Regressive Integrated Moving Average (ARIMA), Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). The analysis unequivocally demonstrates that Transformers outperform the other methods in the chosen experiment.

Keywords

Deep Learning, Transformers, Forecasting, Time Series

1. Introduction

In recent years, the rapid progress of information technology has significantly advanced artificial intelligence (AI), especially machine learning and deep learning, transforming problem-solving and obtaining results that were previously unachievable. The innovations brought by the development of AI have also been applied in the context of official statistics. In time series analysis, in addition to traditional statistical methods such as ARIMA, deep neural network models such as LSTM and GRU have proven to be particularly effective in capturing long-term dependencies in sequential data.

Transformers models, initially developed for Natural Language Processing (NLP), are now also being used in time series analysis. These models eliminate the need for recurrent architectures, relying on attention mechanisms to capture contextual relationships, extending their applications beyond NLP. This paper analyses Transformers applied to the time series forecasting of relevant Istat's socio-economic indicators, such as Gross Domestic Product (GDP), unemployment rate, inflation (CPI) and consumer confidence index. The results obtained with Transformers will be compared with those obtained with the traditional ARIMA, LSTM and GRU techniques for each of the mentioned indicators. In the following paragraphs, we will first provide an overview of the related work and then describe the

Ital-IA 2024: 4th National Conference on Artificial Intelligence, organized by CINI, May 29-30, 2024, Naples, Italy
* Corresponding author.
† These authors contributed equally.
angela.pappagallo@istat.it (A. Pappagallo);
al.emanuele@reply.it (A. Emanuele);
francesco.pugliese@istat.it (F. Pugliese)
massimo.decubellis@istat.it (M. De Cubellis)

methods applied, results obtained and draw conclusions.

2. Related Works

Temporal data is ubiquitous in today's data-driven world. Time Series Forecasting (TSF) [1] is a long-standing task with a wide range of applications. Over the past decades, TSF solutions have evolved from traditional statistical methods (such as ARIMA [2]) to deep learning-based solutions such as Recurrent Neural Networks (RNNs) [3] and Temporal Convolutional Networks (TCNs) [4]. Another commonly employed method is Exponential Smoothing, including variants such as Holt-Winters seasonal method [5], which is effective for capturing trends and seasonality in time series data. Recently, there has been a surge in Transformer-based solutions for time series analysis, as highlighted in [6]. The main strength of transformers lies in their multi-head self-attention mechanism, which has a remarkable ability to extract semantic correlations between elements in a long sequence. However, although the use of different positional encoding techniques can preserve some information about the order, there is still an inevitable loss of temporal information after the self-attention mechanism is applied. This is usually not a serious problem for semantically rich applications such as NLP [7], where the semantic meaning of a sentence is largely preserved even if some words are reordered [8]. However, in time series analysis, the semantic context of the numerical data itself is typically absent, and we are primarily interested in modelling temporal changes between a continuous set of points [9]. In other words, the order itself plays the most important role. Consequently, some researchers have asked the following question: Are transformers really effective for long-term forecasting of time series? [10].

3. Methods

As mentioned in the previous sections, the aim of the following paper is to demonstrate the effectiveness of Transformer models for time series forecasting. In order to do this, a comparative analysis of these models has been carried out with methods that have been the state of the art in time series forecasting for many years. Specifically, the performance of a Transformer model has been compared with that of three other models: Long Short-Term Memory (LSTM) model, a Gated Recurrent Unit (GRU) and a PROPHET model. LSTM networks [11] are a specialised type of Recurrent Neural Networks (RNNs) tailored to the challenge of capturing long-term dependencies. While RNNs excel at using past information for current tasks, they can struggle when the time gap between relevant information and its application is substantial. Although they share a general structure with RNNs, LSTMs have a distinct architecture in their repetition module. Unlike RNNs, which typically consist of a single neural network layer, LSTMs have four interconnected layers. A central component of LSTMs is the cell state (C) which persists throughout the chain and is modified by gate mechanisms. These gates facilitate the selective retention or addition of information to the cell state, thereby enhancing the network's ability to capture and maintain long-term dependencies. On the other side, the GRU [12] can be seen as a simplification of the LSTM where explicit cell states are not used. Another difference is that the LSTM directly controls the flow of information exchanged in the hidden state using separate forget and output gates. Instead, a GRU uses a single reset gate to achieve the same goal. However, the underlying idea of Gated Recurrent Units is quite similar to that of LSTMs in terms of how hidden states are partially reset. Just as LSTM uses input, output and forget gates to decide how much information to carry over from the previous time step to the next, GRU uses update and reset gates. GRU has no separate internal memory and also needs fewer gates to perform the update from one hidden state to another. This raises the question of the specific function of the update and reset gates. The reset gate determines the amount of the hidden state to transfer from the previous time step for a matrix-based update, such as an RNN. The update gate determines the 'relative strength' of the contributions from this matrix-based update and a more direct contribution from the hidden vector to the previous time step. By allowing a direct (partial) copy of hidden states from the previous level, the gradient flow becomes more stable during backpropagation. The update gate simultaneously serves as input and forget gates in LSTMs. Although GRU is a related simplification of LSTMs, it should not be considered a special case of them. Research has shown that the two models perform similarly, with relative performance depending on the task. GRU is easier to implement and more efficient. It may generalize slightly better with less data due to fewer parameters, while LSTM would be preferable with a larger amount of data. PROPHET, which was developed by Facebook (Meta) in 2017 [13], is a time series forecasting model that is specifically designed to handle the common characteristics of economic time series. It is important to note that the model was designed with intuitive

parameters that can be adjusted without requiring knowledge of the underlying model details. This allows analysts to effectively tune the model. The model uses a decomposable time series model consisting of three main components, which are combined additively, like the ARIMA model. These components are trend, seasonality, and holidays. The first two components have already been encountered in the ARIMA model, while the third component, holidays, represents the effects of holidays that occur at potentially irregular intervals over one or more days. In this model, only time is used as a regressor. The problem of forecasting is approached as a curve fitting exercise, which is fundamentally distinct from time series models that explicitly consider the temporal dependence structure of the data. Although this formulation sacrifices some important inferential benefits of a generative model like ARIMA, it offers several practical advantages. It can easily accommodate seasonality with multiple periods and enable the analyst to make different assumptions about trends. PROPHET is capable of handling multiple cases and does not require regularly spaced measurements like ARIMA models, which are designed specifically for forecasting univariate time series. No changes in content have been made. The fitting process is fast and allows for interactive exploration of various model specifications. Finally, the Transformers are a type of machine learning model that was introduced in 2017 by Google's researchers [14]. They are an artificial neural network designed to process sequences of data, such as words in text. Transformers differ from other neural network architectures such as RNNs in that they rely on self-attention rather than recurrence functions. Self-attention allows for relevant parts of an input sequence to be given more weight during the processing of a specific data instance. This enables Transformers to process information in parallel, rather than sequentially, unlike RNNs. The Transformer model has been successfully applied in various natural language processing tasks, including automatic translation, text summarization, text generation, and speech recognition. Its architecture consists of an encoder-decoder structure. The encoder maps a sequence of input symbolic representations to a sequence of continuous representations. The decoder generates an output sequence one element at a time based on these inner representations. The model is autoregressive at each step, using previously generated values as additional input for generating the next ones. The Transformer implements this architecture using multiple self-attention layers and fully connected point-wise layers for both the encoder and the decoder. An attention function maps a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The query vector is derived from the input sequence and is used to determine which parts of the sequence are relevant for the current token. The query vector represents the current input token or element in the sequence, while the key vector represents the context or information from other tokens in the sequence. Similarly, the key vector is also derived from the input sequence and serves as a reference for comparing against the query vector to determine the relevance of each token in the sequence. The value vector represents the information or content associated with each token in the sequence. It is derived from the input sequence and provides the actual representation of each token, like the query and key vectors. The output is determined by a weighted sum of the values. The weight assigned to each value is calculated from a query's compatibility function with the corresponding key. The input comprises queries and keys of size dk and values of size dv. The dot product of the query with all keys is calculated, each divided by $\sqrt{dk}$, and a SoftMax function is applied to obtain weights on the values. The attention function is calculated on a matrix Q, which contains a set of queries. Matrices K and V contain the keys and values, respectively, which are also grouped together. This ensures a simultaneous calculation of the attention function, and so this process is fully parallelizable.

4. Experiment

In this experimentation we prove Transformer models' effectiveness in time series forecasting by comparing them with three other established models like PROPHET, GRU, and LSTM.

4.1. Datasets and Pre-processing

The experiment analised four socio-economic time series collected from Istat. Gross Domestic Product (GDP) measures a country's economic activity over a period, usually a quarter or year. The Unemployment Rate reflects labour market conditions, with increases signaling economic contraction and decreases indicating recovery. Inflation reflects continuous price increases and is crucial for assessing consumer purchasing power. The Consumer Confidence Index measures public economic sentiment, which influences spending and investment decisions and often predicts future economic trends, guiding policy. The GDP data covers the period from March 1, 1990, to March 1, 2023, on a quarterly basis. The

unemployment data range from March 1, 2004, to March 1, 2023, on a quarterly basis. Inflation data range from January 1, 1997, to June 1, 2023, on a monthly basis. The Consumer Confidence Index data range from January 1, 1998, to May 1, 2023, on a monthly basis.

Before entering the data into the Transformer, a pre-processing phase was carried out. This involved analysing the data, including cleaning, transforming, and preparing the raw data to make it suitable for further analysis or for use in machine learning models and algorithms. This phase is crucial because real data can be dirty, incomplete, or in formats unsuitable for analysis. Pre-processing aims to make the data more accurate, consistent, and usable. In the experiment at hand, the pre-processing techniques used for the available data essentially consist of three steps: interpolation to convert quarterly series into monthly series to increase the sample size, data normalization, and a final transformation of the series to a format suitable for supervised learning.

4.2. Results

This section presents the results obtained by the PROPHET, GRU, LSTM, and Transformer models on GDP, Inflation, Consumer Confidence Index, and Unemployment Rate data. The models' performance was evaluated using Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and R^2 (Coefficient of Determination) metrics. RMSE is a measure of the dispersion between observed values and values predicted by a model. MAE calculates the average of the absolute differences between the predicted and observed values. R^2 is a measure of how well a regression model fits the data.

Table 1 displays the metrics calculated on the denormalised GDP dataset.

Table 1
Metrics calculated on GDP dataset.

Model	RMSE	MAE	R^2
PROPHET	15899.18	9208.55	-0.003
GRU	11207.66	7490.36	0.775
LSTM	10492.32	6611.40	0.803
Transformer	**4080.38**	**2296.78**	**0.970**

The metrics indicate that all the models perform reasonably well on the GDP dataset, with the Transformer architecture performing the best while LSTM has comparable performance to that of GRU. However, the PROPHET model performs poorly on the test data as its errors exceed those generated by the mean. These results suggest that the attention mechanism, along with the introduction of temporal encoding, offers significant advantages over standard methods, such as recurrence. Figure 1 shows the predictions of the Transformer, LSTM, and GRU models on the GDP test set. All predictions capture the trend of the series, with some deviations, particularly in the case of LSTM and GRU. The Transformer model's prediction is precise and captures the local maxima and minima of the series effectively. It is important to note that the test dataset includes the period of the Covid-19 pandemic, which is identifiable in the depression exhibited by the curve in the figure. Despite being an unpredictable and anomalous event, the Transformer model manages to capture the trend of the curve better than the two neural networks, especially in the period following the pandemic, albeit with a slight delay. Although LSTM and GRU models lose their predictive capability after the Covid-19 period, the Transformer continues to accurately capture the trend of the series.

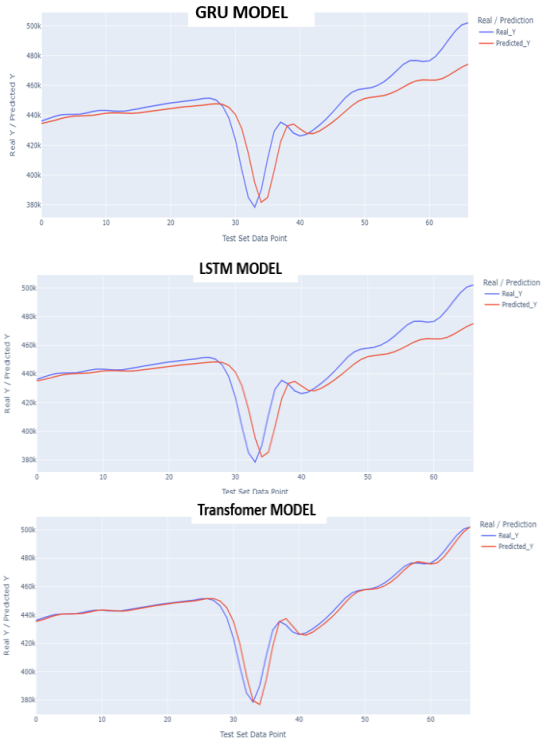

Figure 1: Forecasting on GDP for GRU, LSTM, and Transformer models.

Table 2, Table 3, and Table 4 display the metrics calculated respectively on the denormalised Unemployment Rate, Inflation and Consumer Confidence index (CCI) datasets.

Table 2
Metrics calculated on Unemployment Rate dataset.

Model	RMSE	MAE	R^2
PROPHET	0.30	0.22	0.07
GRU	0.26	0.19	0.91
LSTM	0.26	0.20	0.91
Transformer	**0.21**	**0.14**	**0.94**

Despite a slightly lower performance compared to the previous series, the transformer model still outperforms the other models based on the unemployment rate dataset. Table 2 shows that the Transformer metrics are comparable to those of LSTM and GRU, although they are better. Figure 2 displays the predictions made by the Transformer, LSTM, and GRU models on the Unemployment Rate test set. This series is identical to the previous one, including the Covid-19 pandemic period in the test dataset. The Transformer model remains the reference model in terms of behavior, particularly when considering the period after the pandemic. The LSTM and GRU models produce less accurate predictions.

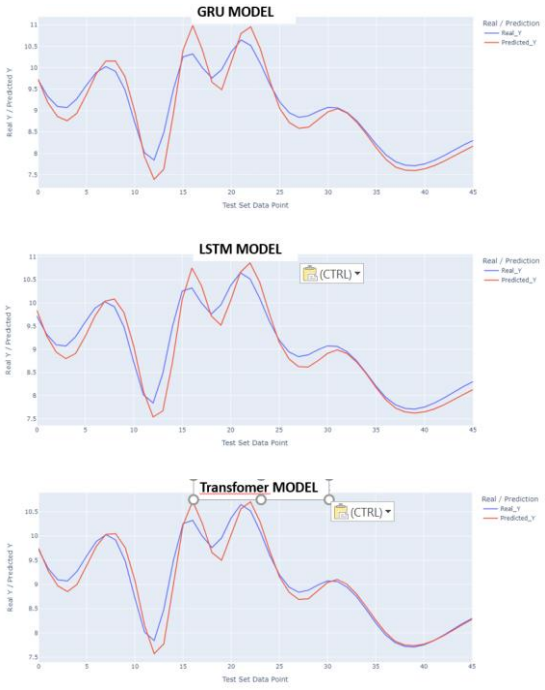

Figure 2: Forecasting on Unemployment Rate for GRU, LSTM, and Transformer models.

Table 3
Metrics calculated on Inflation dataset.

Model	RMSE	MAE	R^2
PROPHET	1.93	0.81	-0.00003
GRU	1.31	0.63	0.88
LSTM	1.33	0.65	0.87
Transformer	**1.22**	**0.59**	**0.89**

Table 4
Metrics calculated on CCI dataset.

Model	RMSE	MAE	R^2
PROPHET	4.62	3.56	-0.001
GRU	3.76	2.87	0.75
LSTM	3.77	2.87	0.74
Transformer	**3.62**	**2.70**	**0.76**

From Table 3 and Table 4, once again, it can be observed that the Transformer model outperforms the other models, albeit in a smaller sample size compared to the previous examples. This could be attributed to the fact that Transformer models perform better with larger amounts of data. With a reduced sample size, the performance of this architecture is closer to that of LSTM and GRU neural networks, but still more efficient at a predictive level. Due to space limitations, we will not display the predictions made on the Inflation and Consumer Confidence Index test sets.

5. Conclusions

In this work, an investigation was conducted on the use of a Transformer-based architecture for Time Series Forecasting (TSF). The architecture has been applied to four different problems, and its performance has been compared with that of three classical TSF models, such as PROPHET, GRU, and LSTM. The results indicate that the Transformer architecture outperforms traditional methods in all experiments, demonstrating its effectiveness in forecasting historical series as well as in the field of Natural Language Processing. However, in some cases, the Transformer's performance approached that of traditional recursive methods. This suggests that, in the TSF domain, the attention mechanism's benefits are more evident when processing high-dimensional data, specifically datasets with a large number of features. The Transformer model performed best on the historical series of GDP, which had the highest number of observations compared to

the other models. Thus, Transformers have been proven to be effective in long-term time series forecasting. However, it is important to emphasize the importance of careful pre-processing and thorough data examination before integrating them into the Transformer model. Additionally, the size of the dataset plays a crucial role in the performance of the Transformer. Specifically, larger datasets tend to produce better results due to the abundance of information available for analysis.

References

[1] Chatfield, Chris. Time-series forecasting. Chapman and Hall/CRC, 2000.

[2] Shumway, Robert H., et al. "ARIMA models." Time series analysis and its applications: with R examples (2017): 75-163.

[3] Hewamalage, Hansika, Christoph Bergmeir, and Kasun Bandara. "Recurrent neural networks for time series forecasting: Current status and future directions." International Journal of Forecasting 37.1 (2021): 388-427.

[4] Wan, Renzhuo, et al. "Multivariate temporal convolutional network: A deep neural networks approach for multivariate time series forecasting." Electronics 8.8 (2019): 876.

[5] Hyndman, R. J., & Athanasopoulos, G. (2018). Forecasting: principles and practice. OTexts.

[6] Wen, Qingsong, et al. "Transformers in time series: A survey." arXiv preprint arXiv:2202.07125 (2022).

[7] Sun, Chi, et al. "How to fine-tune bert for text classification?." Chinese computational linguistics: 18th China national conference, CCL 2019, Kunming, China, October 18–20, 2019, proceedings 18. Springer International Publishing, 2019.

[8] Touvron, Hugo, et al. "Llama 2: Open foundation and fine-tuned chat models." arXiv preprint arXiv:2307.09288 (2023).

[9] Boussif, Oussama, et al. "Improving* day-ahead* Solar Irradiance Time Series Forecasting by Leveraging Spatio-Temporal Context." Advances in Neural Information Processing Systems 36 (2024).

[10] Zeng, Ailing, et al. "Are transformers effective for time series forecasting?." Proceedings of the AAAI conference on artificial intelligence. Vol. 37. No. 9. 2023.

[11] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.

[12] Cho, Kyunghyun, et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." arXiv preprint arXiv:1406.1078 (2014).

[13] Taylor, Sean J., and Benjamin Letham. "Forecasting at scale." The American Statistician 72.1 (2018): 37-45.

[14] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).