The European Journal of Technical and Natural Sciences 2025, No 6
ISSN 2414-2352

PREMIER Publishing
ppublishing.org

DOI:10.29013/EJTNS-25-6-158-162

CC BY

FORECASTING STATE MACROECONOMIC INDICATORS WITH ARTIFICIAL INTELLIGENCE TOOLS

Zurab Tuskia [1]

[1] Georgian Technical University

Cite: *Tuskia, Z. (2023). Forecasting State Macroeconomic Indicators with Artificial Intelligence Tools. European Journal of Technical and Natural Sciences 2025, No 6*. https://doi.org/10.29013/EJTNS-25-6-158-162

Abstract

In an era of rapid advances in artificial intelligence (AI), innovation increasingly shapes many domains of human activity. Public administration is no exception: across the world, governments are adopting AI for forecasting and planning-from the Baltic states' e-Governance platforms to Southeast Asia's "smart state" initiatives (e.g., Singapore) and national AI programs in the Middle East. The shared goal is to raise the quality of forecasts and to make planning, crisis management, and economic policy more responsive to change.

This paper presents a practical, step-by-step methodology for producing targeted, one-year forecasts of state macroeconomic indicators. The approach combines time-series analysis and neural networks with deep learning (BiLSTM + Attention), using efficient, iterative procedures that systematically increase forecast accuracy.

Kywords: *Time Series; Deep Learning (BiLSTM-Attention); ETS; ARIMA; Deep Learning*

Methodology

Data – We use primary data from Georgia's National Statistics Office covering 17 key macroeconomic indicators. For this study, we selected eight: GDP, GDP per capita, imports, state revenues, broad money (M3), average USD/GEL exchange rate, number of employees, and agricultural production output.

$$x_f^{\min} = \min_t x_{t,f}, \quad x_f^{\max} = \max_t x_{t,f},$$

$$R_f = \max(x_f^{\max} - x_f^{\min}, \varepsilon)$$

$$\tilde{x}_{t,f} = \frac{x_{t,f} - x_f^{\min}}{R_f} \in [0,1],$$

The sample spans 2008–2023 with quarterly observations (Q1–Q4). The task was to produce a 2024 forecast achieving an average accuracy above 90% for the chosen indicators, F=Ffull={All quarters 2008–2023}. All 2008–2023 quarterly data were preprocessed using Min–Max normalization, then split into TRAIN (2008–2022) and TARGET (2023). We formed sliding windows and targets for a four-quarter horizon: the previous four quarters as inputs and the subsequent four as targets (for each admissible index within TRAIN).

$$X_i = \left[\tilde{z}_i, \tilde{z}_{i+1}, \ldots, \tilde{z}_{i+W-1}\right] \in \mathbb{R}^{W \times F}.$$

First pass (global model) – We remove a strong stationary component via a naïve "hold-last" baseline and train a global, multi-output model on the residuals. We then construct a *teacher* for future quarters and run quarter-level models with warm starts. In brief: Full matrix, normalized by Min–Max-(2008–2023). Window width W, last index $t = i + W - 1$

$$\hat{Z} \in [0,1]^{T\times P}, \ \text{naive}_{i,h,k} = \tilde{z}_{i+W-1}, p_k.$$

$$\text{resid}_{i,h,k} = Y_i[h,k] - \text{naive}_{i,h,k}$$

Baseline & residuals – The "hold-last" (naive) forecast serves as a baseline; the model learns to predict residuals. p_k – **colum index in** matrix $\hat{Z}$.

Deep model (BiLSTM + Attention, multi-output)

$$e_t = V^T \tanh(W\, h_t), \quad \alpha_t = \frac{\exp(e_t)}{\sum_{j=1}^{W} \exp(e_j)},$$

$$c = \sum_{t=1}^{W} \alpha_t h_t.$$

Encoder: BiLSTM(64) → Dropout(0.2) → BiLSTM(32, return_sequences) Head: Dense(64, ReLU) → Dropout(0.2) → Dense → reshape (multi-horizon outputs). Loss: horizon-weighted MAE.

$$L = \frac{1}{N}\sum_i \frac{1}{HT}\sum_{h=0}^{H-1}\sum_{k=1}^{T} w_h \left|\widehat{\text{resid}}_{i,h,k} - \text{resid}_{i,h,k}\right|,$$

$$\sum_h w_h = 1.$$

Ensembling-Train multiple replicas with different random seeds and average the predictions.

$$\widehat{\text{resid}}_{h,k}^{(\text{ens})} = \frac{1}{M}\sum_{m=1}^{M} \widehat{\text{resid}}_{i,h,k}^{(m)}.$$

$$\widehat{\text{resid}}_{h,k}^{\text{future}} = \tilde{z}_{\text{last}}, p_k$$

Seasonal profile & teacher (grid-search)- We assume a seasonal profile based on average quarterly values on TRAIN; combine it with the residual model to create a stable *teacher* for the TARGET quarters.

Composition – Choose among candidate compositions by minimizing MAE on TRAIN windows. **Warm-start quarterly heads –** Transfer the encoder to compact scalar heads (per target and quarter) and calibrate only the output bias to align with the teacher at initialization.

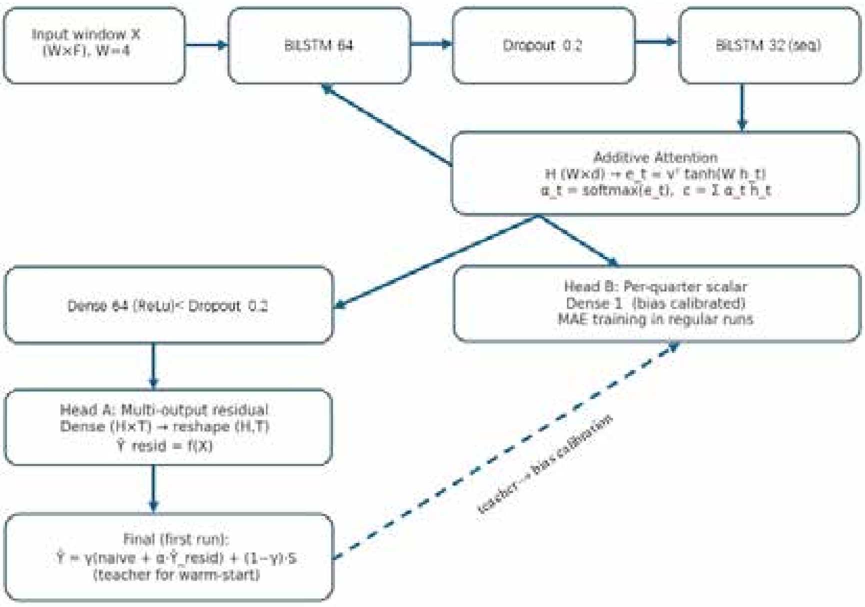
$$Y_{h,k}^{\text{tech}} = \gamma\left(\text{native}_{h,k}^{\text{future}} + \alpha \cdot \widehat{\text{resid}}_{h,k}^{(\text{ens})}\right) +$$

$$+ (1-\gamma) S[h,k],$$

$$\hat{y}_{k,q}(X_{\text{future}}) = \hat{Y}_{q,k}^{\text{teach}}.$$

The global model learns residual structure; blending with the seasonal component stabilizes the teacher; quarterly heads start from a well-calibrated bias.

Figure 1. *DL Model (BiLSTM + Attention)*



Regular reruns with feedback and a nudge-controller – For each (target, quarter) pair:

- If feedback is **GOOD**, the quarter's head is *locked* and excluded from further training.
- If **BADUP/BADDOWN**, perform local adaptation with controlled step size and a trust region.

$$\mathrm{acc} = 1 - \left|\hat{y} - y\right|.$$

Feedback classification – Threshold θ is set to 0.99 (i.e., 99%).

$$\mathrm{GOOD} \Leftrightarrow \mathrm{acc} \geq \theta, \quad \mathrm{BADUP} \Leftrightarrow \hat{y} < y,$$

$$\mathrm{BADDOWN} \Leftrightarrow \hat{y} > y,$$

Local reinforcement- For BADUP/BADDOWN, increase the weight of the last TRAIN window and perform a small, bounded update.

$$y^*_{last} = \begin{cases} y_{last}(1+\delta), & \mathrm{BADUP} \\ y_{last}(1-\delta), & \mathrm{BADDOWN} \end{cases},$$

$$\delta = \mathrm{FEEDBACK\_SHIFT},$$

Bounded nudge without TARGET leakage – Use an adaptive base push; a seasonal *anchor*; a minimal step; and a trust corridor around the planned target. In the late *POLISH* phase, updates are one-sided (BADUP only upward; BADDOWN only downward).

$$\mathrm{BADUP}: p_{\mathrm{nudget}} = p + k(1-p),$$

$$\mathrm{BADDOWN}: p_{\mathrm{nudged}} = p - kp,$$

$$p_{\mathrm{anch}} = (1-\lambda)p_{\mathrm{nudged}} + \lambda S_q, \quad \lambda \in [0,1].$$

Using the controlled sequence of steps "push → anchor → minimal step → confidence corridor," and **without** training on the current TARGET year's data (2023 in this case), we achieved **GOOD** values of the feedback metric for all quarters and saved the stable models into a single model file. Using the stored quarterly weights ("GOOD-locks"), we built the forecast for the next year (2024 in this case) in **RAW** units. We formed X_{future} from the last $W = 4$ quarters of the full history (TRAIN + TARGET). For each target k and quarter q, we loaded the weights and computed the normalized forecast $\hat{y}^{\mathrm{norm}}_{(k,q)}$. Transition to the RAW space: $\hat{y}^{\mathrm{raw}}_{(k,q)} = \hat{y}^{\mathrm{norm}}_{(k,q)} \cdot R_{p_k} + x^{\min}_{p_k}$, where R_{p_k} and $x^{\min}_{p_k}$ are the min–max normalization parameters for the corresponding target column p_k.

Results

With the proposed methodology, the **average forecast accuracy for 2024 reached 93.22%**. To benchmark, we uploaded the same macro data to emulated models on several platforms (ChatGPT, Amazon AWS, Microsoft Azure) to obtain their 2024 forecasts. We then compared quarterly accuracies per indicator and the overall average:

Feature	Year	Quarter	Forecast Accuracy GPT CHAT %	Forecast Accuracy AMAZON AWS %	Forecast Accuracy MS AZURE %	Forecast Accuracy Our Project %
GDP (MILLION $)	2024	I	73.61	92.01	92.01	97.88
GDP (MILLION $)	2024	II	71.27	91.37	91.37	97.37
GDP PER CAPITA $	2024	IV	70.15	89.85	89.85	90.57
IMPORTS (MILLION $)	2024	I	82.23	98.58	98.58	92.02
IMPORTS (MILLION $)	2024	II	73.37	92.21	92.21	97.44
IMPORTS (MILLION $)	2024	III	73.09	92.15	92.15	95.57

Feature	Year	Quarter	Forecast Accuracy GPT CHAT %	Forecast Accuracy AMAZON AWS %	Forecast Accuracy MS AZURE %	Forecast Accuracy Our Project %
IMPORTS (MILLION $)	2024	IV	71.70	84.23	84.23	85.33
STATE REVENUES (MILLION $)	2024	I	60.12	82.50	82.50	84.32
STATE REVENUES (MILLION $)	2024	II	63.36	92.03	92.03	97.36
STATE REVENUES (MILLION $)	2024	III	66.22	88.55	88.55	92.45
BROAD MONEY M3 (MILLION $)	2024	IV	72.68	89.64	89.64	90.56
AVERAGE EXCHANGE RATE USD-$	2024	I	82.75	95.88	97.81	92.25
AVERAGE EXCHANGE RATE USD-$	2024	IV	84.19	97.46	98.06	98.99
NUMBER OF EMPLOYEES	2024	I	96.18	95.56	95.56	98.21
NUMBER OF EMPLOYEES	2024	II	96.98	95.82	95.82	96.85
NUMBER OF EMPLOYEES	2024	III	98.17	96.14	96.14	97.10
NUMBER OF EMPLOYEES	2024	IV	99.48	96.36	96.36	96.72
AGRICULTURAL PRODUCTION OUTPUT (MILLION $)	2024	I	86.54	95.90	95.89	89.29
AGRICULTURAL PRODUCTION OUTPUT (MILLION $)	2024	IV	83.96	90.28	90.28	95.45
Average Forecast Accuracy			77.83	91.94	92.04	93.22

This methodology blends several practices known from classical modeling, but using them as a unified ensemble yields distinctive advantages:

1. ***Two-stage architecture ("global encoder → quarterly heads").***

Rather than training "one big network," we train a single multi-task (BiLSTM + Attention) model on residuals, transfer its encoder into small quarterly heads (per target and quarter), and then perform calibration with reinforcement-style updates.

2. ***Controlled, monotonic correction without data leakage.***

In reruns we do *not* feed TARGET-year data to the network. Instead, we adjust only the output bias of the quarterly head with small, controlled steps toward the planned point. Modes (TURBO/AGGR/POLISH) adapt automatically based on the shortfall to the GOOD threshold; the nudge intensity and trust interval width are adjusted accordingly. Once a (target, quarter) crosses the threshold, we lock that head, preventing later degradation.

3. ***Micro-models ("quarter × target") instead of a monolithic head.***

Quarters behave differently in practice. Splitting heads by quarter yields natural localization and faster convergence to accurate forecasts.

Conclusion

In today's environment-where economic stability and well-targeted investment decisions are prerequisites for national development-forecasting plays a crucial role. Using macroeconomic data from 2008–2023, we set out to build a model capable of rationally predicting 2024 indicators (we chose 2024 specifically to allow straightforward accuracy verification). The challenge required not only capturing historical trends but also uncovering the internal logic and interrelations among those trends. Our streamlined, practical pipeline couples time-series analysis with a neural baseline and then refines it through carefully controlled deep-learning adjustments, reaching 90–95% accuracy. The resulting next-year forecasts achieved **93.22%** average accuracy in this study-useful for planning and for optimizing the management of public resources.

References

Hyndman & Athanasopoulos (2021) "Forecasting: Principles and Practice (FPP3)".
Defu Cao, Yixiang Zheng, Parisa Hassanzadeh… (2023) "Large Scale Financial Time Series Forecasting with Multi-faceted Model".
– Bahdanau, Cho, Bengio (2014) "Neural Machine Translation by Jointly Learning to Align and Translate".
Hyndman, R. J., Koehler, A. B., Ord, J. K., & Snyder, R. D. (2002) A state space framework for automatic forecasting using exponential smoothing.
Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015) Time Series Analysis: Forecasting and Control.

submitted 15.11.2025;
accepted for publication 29.11.2025;
published 30.12.2025
© Tuskia, Z.
Contact: zurabtuskia@gmail.com