Using Hybrid Deep Neural Network to Forecast Macroeconomic Indicators

ZURAB BOSIKASHVILI, ZURAB TUSKIA, TALIKO ZHVANIA
Faculty of Informatics and Control Systems,
Georgian Technical University,
77, Kostava Str., Tbilisi, 0160,
GEORGIA

Abstract: - Economic policymaking in complex, evolving systems relies upon accurate forecasts of economic conditions. To address this challenge, the paper presents a comprehensive and systematic evaluation of a previously proposed deep neural network (NN) architecture. Based on this evaluation, an advanced hybrid model composed of Long Short-Term Memory (LSTM), attention, and nudging layers is proposed. To enhance the accuracy of forecasting macroeconomic indicators, an adaptive algorithm was developed to correct forecast values within the framework of the hybrid NN architecture. This algorithm gradually improves predictions through nudges based on the data assimilation (DA) method. Both synthetic and real data were used for experiments on Georgia's budget forecasting, demonstrating high forecasting accuracy without requiring extensive domain knowledge and robustness to changes in certain system parameters.

Key-Words: - hybrid model; data assimilation; attention; nudging, deep learning; neural networks; macroeconomic indicator;

Received: June 2, 2026. Revised: August 7, 2026. Accepted: September 11, 2026. Published: November 26, 2026.

1 Introduction

The rapid advancement of machine learning, especially through deep neural networks, has greatly enhanced our ability to forecast dynamic processes for next-generation applications in fields such as macroeconomic forecasting, energy consumption, budget forecasting, financial time series, and more. This progress has integrated both structural and non-structural approaches to forecasting. The structural approach is based on economic theory and aims to create empirical forecasts. In contrast, the non-structural approach involves selecting models or methods without directly referencing a foundational economic model. In addition to forecasting, deep neural network models can identify dependencies and patterns among various features. Moreover, when explanatory models are included, they effectively combine elements from both structural and non-structural methods.

Formally, the task of non-structural forecasting of macroeconomic indicators is a multivariate multi-step forecasting task of time series, the approaches and methods of which are discussed in [1], [2], [3], [4]. Budget and financial forecasting research can be grouped into six methodological streams.

First, classical statistical models (ARIMA/ SARIMA/ exponential smoothing families) remain the standard baseline in applied forecasting because they are transparent, operationally simple, and easy to audit in institutional settings. Their strengths: highly interpretable (easy to explain to finance & auditors), work well with stable, seasonal historical data and are reliable baseline models.

Weaknesses: struggle with structural breaks (policy changes, crises) and are poor at modeling nonlinear relationships

Second, machine-learning models improve flexibility when many predictors are available. Random Forests leverage diversified tree ensembles; Elastic Net is effective in high-dimensional and collinear settings; and gradient-boosted trees (e.g., XGBoost) are widely used because they combine strong predictive performance with scalable training. The trade-off is that these models usually require careful feature design and stricter controls against overfitting than classical baselines. Strengths: capture nonlinear relationships and interactions, handle many explanatory variables (inflation, GDP, enrollments, etc.), often outperform classical models in medium-complexity settings, feature importance helps partial interpretability. Weaknesses: require careful feature engineering, less transparent than statistical models, risk of overfitting if data history is short.

Third, deep learning approaches address complex temporal structure. Recurrent and attention-based architectures have shown strong results for sequence modeling, while forecasting-specific variants such as DeepAR and TFT highlight the value of learning

across many related series and supporting multi-1. horizon prediction. In practice, however, governance2. and transparency requirements, especially in public-3. finance contexts, still constrain full replacement of interpretable pipelines by black-box systems. Strengths: excellent at complex temporal dynamics, can model long-term dependencies and irregular patterns, adapt well to large, high-frequency datasets, useful for scenario-based forecasting. Weaknesses: low interpretability ("black box"), high data and computational requirements, hard to justify forecasts to stakeholders, sensitive to data quality and regime changes.

Fourth, reinforcement learning (RL) is primarily positioned as a sequential decision framework (allocation/policy optimization under constraints), not as a direct substitute for point forecasting. Literature reports promise dynamic control problems, but also recurring challenges in robustness, explainability, and validation. Strengths: optimize decisions, not just forecasts, adapt dynamically to changing environments, suitable for multi-year budget planning under constraints. Weaknesses: not a pure forecasting method, requires simulation environments, difficult to validate and explain, high implementation complexity.

Fifth, Nudging combined with deep learning leverages behavioral insights and AI to guide decisions effectively. Strengths: include personalized interventions, real-time feedback, and improved learner outcomes, especially in education and health. Weaknesses: include limited long-term empowerment, potential over-reliance on algorithms, and challenges with transparency and user autonomy.

Finally, hybrid and ensemble strategies are consistently supported by forecasting evidence. The ML findings are especially relevant: method combinations were dominant, while pure ML entries were generally weaker than leading combination benchmarks. This motivates mixed designs that prioritize robustness under heterogeneity and regime instability. Strengths: more robust to uncertainty and regime changes, combine interpretability with accuracy, reduce model-specific bias, often best real-world performance. Weaknesses: higher system complexity, more difficult to maintain, require strong evaluation discipline.

To solve the problems of forecasting macroeconomic indicators, we propose a non-structural approach based on an advanced hybrid deep NN, which combines architectural solutions such as Long Short-Term Memory (LSTM) recurrent NN, attention-based NN, and a data assimilation approach (Nudge NN) in a single-layer network. The model we propose provides:

Good study of temporal dependencies;
Finding important time periods;
Targeted correction of the forecast;This achieves high performance while enabling new forms of interpretability. To obtain significant performance improvements over state-of-the-art benchmarks, we introduce multiple novel ideas to align the architecture with the full range of potential inputs and temporal relationships common to multi-horizon forecasting.

2 Related Works

A hybrid LSTM-Attention model combines the sequential memory capabilities of LSTM networks with the ability of attention mechanisms to focus on relevant input features [5].

The hybrid LSTM–Attention model effectively handles missing data in multivariate environmental time series by combining LSTM's sequential learning with attention mechanisms to focus on relevant temporal features [6], [7], [8], [9]. Evaluated on soil temperature and meteorological data (precipitation, temperature, wind speed), it demonstrates superior imputation accuracy compared to traditional methods. The 'sandwiched' architecture enhances feature representation by integrating attention within LSTM layers, improving performance under irregular missing patterns common in environmental monitoring. Hybrid models in financial forecasting are discussed in the paper [10], which enhance accuracy by integrating econometric methods (e.g., ARIMA, ARFIMA) with machine learning and deep learning techniques. The research demonstrates that such combinations improve predictive performance, especially in capturing nonlinearities and volatility clustering in financial time series. Paper [11] proposes a deep learning-based SDGs achievement progress time series prediction framework that effectively addresses the shortcomings of traditional methods in handling high-dimensional nonlinear time series data by integrating LSTM networks and self-attention mechanisms. Experimental results demonstrate that the framework achieves excellent performance in prediction tasks for 17 SDGs targets, with an average prediction accuracy of 92.847%. The study from [12] demonstrates an efficient, resource-aware Attentional Bi-LSTM model deployed on the NanoPi Neo Plus2 for real-time multivariate time series forecasting [12]. By leveraging edge computing, the system enables low-latency, on-device prediction of hazardous trends without relying on cloud infrastructure. The attention mechanism enhances focus on relevant temporal features, improving accuracy while maintaining computational efficiency. This approach is

particularly valuable for applications in industrial monitoring, environmental sensing, and smart systems where responsiveness and privacy are critical.

Either prediction, particularly for probabilistic forecasting where uncertainty quantification is critical.

The "nudging method" in time-series forecasting usually appears in the literature under data assimilation or nudging-based feedback control. The nudging approach in time-series forecasting works by continuously correcting model predictions toward observed data.

The literature has evolved through many stages; we are focusing only on Machine-learning-assisted nudging. Spectral nudging ensures large-scale atmospheric patterns from the physics model are preserved in machine learning outputs, improving consistency and reliability [13], [14]. This approach leverages the strengths of both systems—robust physical laws and data-driven adaptability—while reducing errors in medium-range forecasts. Recent papers combine machine learning with data assimilation. Among them: Data Assimilation with Deep Neural Nets Informed by Nudging [15]. The main idea is to train a deep neural network to approximate the nudging algorithm. Advantage: once trained, the model predicts much faster than solving differential equations. This approach is promising for real-time forecasting. Another stream of research uses nudging in operational forecasting systems. For example, the Nudging-ExRT Data Assimilation Framework generates innovation vectors from observations that are used to train machine-learning models to adjust emissions and improve air-quality forecasting accuracy [16].

While the integration of deep NN and DA has advanced the modeling of nonlinear dynamic systems, significant challenges remain, including handling high-dimensional systems, sparse observations, dynamic noise, and physical constraints [17]. Existing LSTM-based DA methods, such as those by [18], improve predictive accuracy in reduced-order models but struggle with capturing both short- and long-term dynamics in high-dimensional, noisy environments. Recent developments, such as the online paleoclimate DA framework proposed by [19], have demonstrated the feasibility of coupling neural networks with physical assimilation systems in real-time. Single loss functions like MSE or MAE often fail to balance state estimation and error correction, limiting robustness under noisy conditions. Additionally, traditional activation functions and feature extraction strategies may not fully address the dynamic requirements of DA, particularly for time-series anomalies.

The integration of machine learning (ML), particularly deep learning (DL), with data assimilation (DA) has transformed meteorological forecasting, environmental modeling, and physics-constrained DL by addressing the limitations of traditional DA methods in complex, nonlinear, and high-dimensional systems [19], [20].

3 Problem Formulation

3.1 The general formal model of a macroeconomic

A general formal macroeconomic model of any field can be represented by the following set of dynamic nonlinear equations:

$$\begin{cases} \boldsymbol{y}_{t+1} = \boldsymbol{\mathcal{F}}(\boldsymbol{y}_t, \boldsymbol{x}_t) & (1) \\ \boldsymbol{z}_t = \boldsymbol{\mathcal{H}}(\boldsymbol{y}_t, \boldsymbol{\xi}_t) & (2) \end{cases}$$

Where $\boldsymbol{y}_t \in \mathbb{R}^{n_y} -$ is a vector of endogenous macroeconomic variables (e.g., output, GDP, inflation,…) at time t, $\boldsymbol{x}_t \in \mathbb{R}^{n_x}$ - represents the exogenous input vector at time t, which is assumed to be related to $\boldsymbol{y}_t$ and those whose values are determined by factors outside the model itself, and can cause shocks, $\boldsymbol{\mathcal{F}}: \mathbb{R}^{n_x+n_y} \to \mathbb{R}^m$ -is a nonlinear implicit operator, $\boldsymbol{\mathcal{H}} -$ is an observation operator, $\boldsymbol{z}_t \in \mathbb{R}^m -$ observable variables, $\boldsymbol{\xi}_t \in \mathbb{R}^m -$ is a noise vector, $t \in [0, t_c]$- is a time-varying variable (year, quarter, month, day) within a defined range.

In complex dynamic systems, such as macroeconomics, the relationship between input and output indicators can be deeper; the magnitude of the output characteristic at a given moment may depend not only on the previous state of the system, but also on the previous state of the previous one, etc. In many estimations, it is also relevant to determine the form of such dependencies. In forecasting tasks, it is also relevant to determine group dependencies in time; for example, an annual indicator consisting of quarterly indicators, how it depends on the indicators of the previous period, and such a model can be called a multivariate time series, multi-step forecasting model:

$$[\tilde{\boldsymbol{y}}_{t+1}\tilde{\boldsymbol{y}}_{t+2},..,\tilde{\boldsymbol{y}}_{t+H}]^T = \boldsymbol{\mathcal{F}}([\boldsymbol{y}_t,\ldots,\boldsymbol{y}_{t-W}]^T,[\boldsymbol{x}_t,..,\boldsymbol{x}_{t-W}]^T)+\boldsymbol{\xi}_t \quad (3)$$

$\tilde{\boldsymbol{y}} -$is an observable target variable, H and W. Accordingly, the dimensions of the forecast horizon and observation window are.

Problems posed in this way are widely used in management, risk assessment, and forecasting, and are well studied and presented in the scientific literature [4], [6]. In the forecasting of macroeconomic indicators, the noise level can be considered small compared to the level of useful

information, and formula (3) can be rewritten as follows:

$$\widetilde{\boldsymbol{Y}}_{t+1} = \boldsymbol{\mathcal{F}}\left(\ddot{\boldsymbol{X}}_{\boldsymbol{t}}\right) + \boldsymbol{\xi}_t \quad (4)$$

where $\widetilde{\boldsymbol{Y}}_t \in \mathbb{R}^{m\times H}$ is a matrix of features of the predictive target output, $\ddot{\boldsymbol{X}}_t = [\boldsymbol{X_t}, \boldsymbol{Y_t}] \in \mathbb{R}^{W\times(n_x+n_y)}$ is the matrix of input variables defined in formula (3). Later, we represent the combination of two variables in a specific direction using a two-dot notation.

The goal of our problem is to predict the values of the output vectors $\widetilde{\boldsymbol{y}}_t$described by the formula (4) and measured on the interval $(t_c < t \le t_c + H)$ based on historical data obtained in the interval $[0, t_c]$, it's clear $W < t_c$.

The implicit nature of the $\boldsymbol{\mathcal{F}}$ operator makes it challenging to solve the problem effectively. To address this issue, the paper proposes a deep neural network model that employs a multi-step adaptive approximation of the $\boldsymbol{\mathcal{F}}$ operator using deep neural networks of a hybrid (LSTM-Attention-Nudging) architecture based on historical data.

3.2 Proposed Methodological Approach

The proposed model includes two submodels: LSTM-Attention and Nudge submodels. First, let's prepare a formal basis for these models.

Let us assume that we have a historical data set $\boldsymbol{D} = \{\ddot{\boldsymbol{x}}_{\boldsymbol{t}}, \widetilde{\boldsymbol{y}}_{t+1} \mid t = \overline{0,N}\}$ of observations of macroeconomic indicators accumulated over time $[0, t_p]$, which is represented as an N-element time series.

Let $\ddot{\boldsymbol{x}}_t = \left[\boldsymbol{x}_{t,}\widetilde{\boldsymbol{y}}_t\right] \in \mathbb{R}^n$, where $n = n_x + n_y$. Here, $\boldsymbol{x}_t$ and $\widetilde{\boldsymbol{y}}_t$ represent the vectors of exogenous and endogenous variables at time t, respectively, and $\widetilde{\boldsymbol{y}}_{t+1}$ is the multivariate vector of targeted observable variables at time t+1. In the case of macroeconomic features, **D** is the training time series, and the task is a multivariate multi-step supervised machine learning task of time series forecasting, which is represented as follows:

Given the multivariate training set **D,** a complete machine learning prediction operator $\widetilde{\boldsymbol{\mathcal{M}}}$ is defined, which approximates the operator $\boldsymbol{\mathcal{F}}$ defined in formula (4) and predicts the system output variables using the following formula:

$$\{\widehat{\boldsymbol{y}}_{t:t+H}\} = \widetilde{\boldsymbol{\mathcal{M}}}(\{\ddot{\boldsymbol{x}}_{t-w:t}\} \mid \boldsymbol{\Theta}), \boldsymbol{t} \in [0, t_p], \quad (5)$$

where $\{\widehat{\boldsymbol{y}}_{t+1:t+H}\}$ is the multi-step time series of the prediction, and $\boldsymbol{\Theta}$ is the vector of model parameters.

The $\widetilde{\boldsymbol{\mathcal{M}}}$ operator's efficiency from formula (5) can be limited, particularly in complex systems with ill-defined component dependencies. To address this, a multi-layer architecture was proposed: the base layer uses an LSTM-Attention (L-A) model for a preliminary solution, and a second layer, Nudge (N), adaptively adjusts the learned parameters via the Nudging method to improve the prediction accuracy of each target indicator.

For clarity, let us introduce notation according to which multivariate operators and variables are represented by three-dotted caps, and univariate ones by one-dotted caps.

Represent an adaptive, multilayer implementation of the $\widetilde{\boldsymbol{\mathcal{M}}}$ operator using a single $\dddot{\boldsymbol{\mathcal{M}}}^{\Theta}$ of the (L-A) model with weights Θ of a deep neural network (DNN) neuron, and with k univariate prediction nudging (N) model operators $\dot{\boldsymbol{\mathcal{M}}}_{\boldsymbol{i}}$ for each target feature $\boldsymbol{i}$. The prediction can be expressed as follows:

$$\hat{y}^{\boldsymbol{i}}_{t+1} = \dot{\boldsymbol{\mathcal{M}}}_{\boldsymbol{i}}(\{\ddot{\boldsymbol{x}}_{t-w:t}\} \mid \boldsymbol{\Theta_i}), \ (i = 0, k) \quad (6)$$

where $\hat{y}^{\boldsymbol{i}}_{t+1} \in \mathbb{R}$, is the prediction of the i-th target feature, and the training set row $\ddot{\boldsymbol{x}}_t = \left[\boldsymbol{x}_{t,}y_t\right] \in \mathbb{R}^n, (n = n_x + 1), \ k = m + q$. The variables m, q represent the target feature and the quartile counts, respectively. $\dot{\boldsymbol{\mathcal{M}}}_{\boldsymbol{i}}$ can be written as the following formula

$$\dot{\boldsymbol{\mathcal{M}}}_i = \dddot{\boldsymbol{\mathcal{M}}} \circ \wp_{\boldsymbol{i}} \circ \boldsymbol{\mathcal{B}}_i \quad (7)$$

Where $\wp_{\boldsymbol{i}}$ is a projection operator that specifies the target feature from the vector of features.

$\boldsymbol{\mathcal{B}}_{\boldsymbol{i}}$- operator reflects the implementation of the model of a specific i-th feature by DNN, which is represented by the L-A-N architecture.

$$\boldsymbol{\mathcal{B}}_{\boldsymbol{i}}^{\boldsymbol{\theta_i}} = \boldsymbol{\mathcal{C}}_i \circ \boldsymbol{\mathcal{N}}_i^{\boldsymbol{\theta_i}} \quad (8)$$

Where $\mathcal{C}_i$ is a calibration operator that calculates the weights (bias) of the $\dot{\boldsymbol{\mathcal{M}}}_{\boldsymbol{i}}$ model with the values calculated by the $\dddot{\mathcal{M}}$ model weights and the $\mathcal{N}_i$ operator:

$$\dot{\theta}_i = (\ddot{\theta}_i + \alpha_i) \text{ where } \alpha_i = \lambda \times \|\delta\| \quad (9)$$

that depends on the condition numbers of data nudging, the numerical model, and the numerical forecasting model, where $\delta = \hat{y}_t - \widetilde{\boldsymbol{y}}_{\mathrm{t}}$ is the error between the predicted feature and its true value. Learned correction

$$y^*_{t+1} = \hat{y}_{t+1} + \boldsymbol{\mathcal{B}}_i^{\theta_i}(\hat{y}_t - \widetilde{\boldsymbol{y}}_{\mathrm{t}}) \quad (10)$$

The creation of new $\boldsymbol{\mathcal{B}}$ models for predicting each i-th feature can be continued until $\delta < \delta_{tresh}$,, or the number of steps is predetermined.

For the first layer of an L-A-N network, the network loss function can be represented as follows:

$$\dddot{\mathcal{L}}(\theta) = \sum_{j=0}^{N-W+1} \left\|\dddot{\boldsymbol{\mathcal{M}}}^{\theta}(\ddot{\boldsymbol{x}}_j) - \widetilde{\boldsymbol{y}}_j\right\| \quad (11)$$

Accordingly, the parameter optimization problem can be represented as follows:

$\ddot{\theta}^* = \underset{\theta}{\operatorname{argmin}}\{\dddot{\mathcal{L}}(\theta)\}$ Here $\|.\|$ is defined using either the L1 norm or the L2 norm.

For $i = 0\ to\ m \times q$: $\boldsymbol{\mathcal{B}}_i$ the parameters $\dot{\theta}_i$ are calculated from the following formula:

$$\dot{\theta}_i^* = \underset{\theta}{\operatorname{argmin}}\{\|\boldsymbol{\mathcal{B}}_i^{\theta}(\ddot{\boldsymbol{x}}_{N-H+1}) - \tilde{\boldsymbol{y}}_t\|\} \quad (12)$$

Where $\dot{\theta}_i^0 = \ddot{\theta}^*$. To increase the accuracy of parameter calculation, the calculation scheme discussed above can be organized into the following multi-step adaptive process:

$\boldsymbol{\mathcal{B}}_i^{k+1 \to \dot{\theta}_i^{k+1}} \to \boldsymbol{\mathcal{B}}_i^{k \to \dot{\theta}_i^k}$, $\dot{\theta}_i^0$ where is defined by formula (9). The process can be limited by the number of steps k_{max} or by the error threshold δ_{tresh}.

4 Problem Solution

For multivariate time series multi-step estimation, a hybrid multilayer, adaptive model was developed based on the methodology discussed above, the architecture of which is shown in Figure 1.Let's consider each layer separately.

4.1 Preprocessing layer

At the start of the first layer step, to account for different feature scales and to ensure numerical stability, each feature vector must be normalized to the range [0, 1] using min-max normalization. The scaling parameters are taken only from the training set to avoid data leakage.

$$x_{scaled} = (x - x_{min})/(x_{max} - x_{min}) \quad (13)$$

At the next step of the layer, the prediction task was formulated as a supervised learning problem using a sliding window approach, which transforms sequential data into a supervised learning problem by creating overlapping input-output pairs. The sliding window approach in time series prediction transforms sequential data into fixed-length input-output pairs by moving a window across the dataset [1]. This method enables models to learn temporal dependencies by using past observations (input) to predict future values (output). It is widely used in regression-based models and enhances forecasting accuracy by capturing complex patterns. Figure 2 depicts training-data construction using a sliding window. As a result, we get two

$X_{train} = \{[x_{0,0}, x_{0,1}, .., x_{0,W-1}], [x_{1,0}, x_{1,1}, .., x_{1,W-1}], ..., [x_{\bar{n},0}, x_{\bar{n},1}, .., x_{\bar{n},W-1}]\}$ and

$Y_{train} = \{[y_{0,0}, y_{0,1}, .., y_{0,H-1}], [y_{1,0}, y_{1,1}, .., y_{1,H-1}], ..., [y_{\bar{n},0}, y_{\bar{n},1}, .., y_{\bar{n},H-1}]\}$ dataset

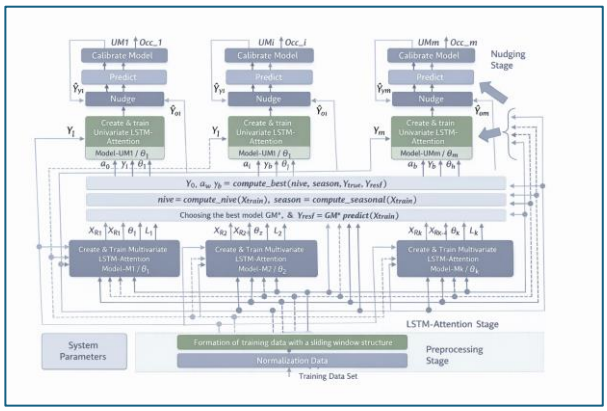


Fig. 1: The LSTM-Attention-Nudging model architecture

Source: created by the authors.

4.2 LSTM-Attention Layer

As can be seen from Figure 1, this layer consists of k LSTM-Attention modules running in parallel. Each module consists of aBiLSTM and an Attention module connected in series. Let's consider them separately.

BiLSTM layer that processes the input sequence to extract rich contextual features from both forward and backward temporal directions. “Bidirectionality” will allow embedding both interpretations in the hidden states.

The construction and use of BiLSTM and LSTM networks have been widely discussed in many works [7], [8]. This is not the focus of our study. Therefore, we will only briefly address their connections with other layers here.

Let's represent this layer as the following transformation of sequences:

$$\tilde{\boldsymbol{h}}_t, \boldsymbol{h}_t, \boldsymbol{c}_t = BiLSTM(\boldsymbol{x}_t, \boldsymbol{h}_{t-1}, \boldsymbol{c}_{t-1}) \quad (14)$$

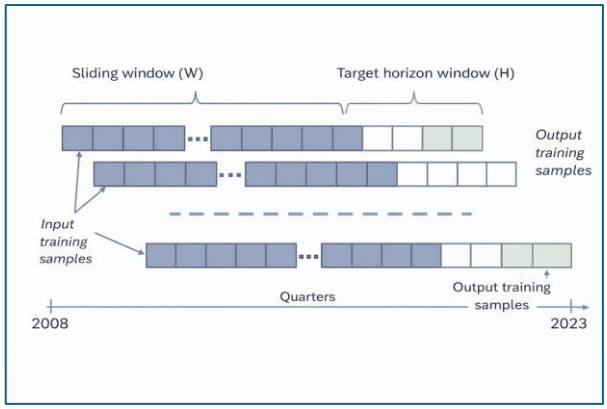


Fig. 2: Illustration of training-data construction using a sliding window: prediction horizon H= 4 quarters and floating window size $\boldsymbol{W}$ = k·$\boldsymbol{T_h}$.

Source: created by the authors.

Where $\boldsymbol{h_t}$ is a hidden state of the $BiLSTM$ module at time t, $\boldsymbol{c_t}$ is a cell state, and $\widetilde{\boldsymbol{h}}_{\boldsymbol{t}}$ is a $\widetilde{\boldsymbol{h}}_{\boldsymbol{t}} = [\boldsymbol{h_{t-w}}, \boldsymbol{h_{t-2}}, \ldots, \boldsymbol{h_{t-1}}]$ the concatenation of all hidden states, which is constructed at w time steps. To better capture complex temporal patterns in historical features, we use two BiLSTM layers (returning sequences), an additive attention layer (which learns temporal importance weights) to produce a context vector, and then use a Dense layer to produce horizon × features outputs.

The attention module determines which moments in time are more important for prediction. To enable the model to dynamically focus on the most relevant time steps, we integrated a Bahdanau-style attention mechanism [9]. The Bahdanau attention mechanism computes alignment scores between the decoder's previous hidden state (query) and all encoder hidden states (keys/values). The attention layer computes a context vector ($\boldsymbol{ctx}$) as a weighted sum of the BiLSTM hidden states. The mechanism calculates alignment scores ($\boldsymbol{e_t}$) which are then normalized via a softmax function to produce attention weights (α_t).

$$\boldsymbol{ctx}_{t_i} = \sum_{t=t_i}^{t_i+W} \alpha_t \, \boldsymbol{h_t} \qquad (15)$$

The score function is calculated from LSTM network parameters $\mathbf{W_h}, \mathbf{b_h}$ and has the following form:

$$\boldsymbol{e_t} = \boldsymbol{v_a}^T tanh(\boldsymbol{W_a h_t} + \boldsymbol{b_a}) \qquad (16)$$

Where $\boldsymbol{v_a}, \boldsymbol{W_a}, \boldsymbol{b_a}$ −learnable weight matrices/vectors.

4. The attention weights α_t are calculated using a feedforward neural network**:**

$$\boldsymbol{\alpha_t} = \frac{exp(\boldsymbol{e_t})}{\sum_{k=1}^{W} exp(\boldsymbol{e_{t_k}})} \qquad (17)$$

This allows the model to dynamically focus on relevant parts of the input sequence and provides a weighted representation of history.

The summary of the LSTM-Attention network is passed to the base prediction sublayer, which is presented as an MLP constructed by Dense nodes.

The final operation is the evaluation of residual and naïve values of the prediction.

The naive baseline simply repeats the last observed vector for each future step (a strong, simple baseline for many real series) and will be presented in the next form:

$$\widetilde{\boldsymbol{y}}^{\boldsymbol{n}}_{\boldsymbol{t-H:t}} = \widetilde{\boldsymbol{y}}_{\boldsymbol{t}} \qquad (18)$$

The residual baseline is calculated as:

$$\widehat{\boldsymbol{y}}^{\boldsymbol{r}}_{\boldsymbol{t-H:t}} = \widehat{\boldsymbol{y}}_{\boldsymbol{t-H:t}} - \widetilde{\boldsymbol{y}}^{\boldsymbol{n}}_{\boldsymbol{t-H:t}} \qquad (19)$$

Where $\widehat{\boldsymbol{y}}_{\boldsymbol{t-H:t}}$ is the target features, and the prediction is performed by the constructed model. As a result, the

The model is trained end-to-end by minimizing the mean absolute error (MAE) loss using the Adam optimizer. A prediction was made for the final training window, and the residual and naive values were evaluated.

But since the optimization result for deep neural networks depends heavily on the choice of initial values of the parameters, in this paper, **ν** number of LSTM-Attention neural layers are selected simultaneously, and the optimization is performed independently on them with randomly selected initial values. Accordingly, instead of one, we get **ν** results.

The pipeline of the i-th branch of the BiLSTM-Attention model looks like this:

$$(X,Y)_{train}, \Theta_0 \rightarrow LSTM - Attention^i \rightarrow (ctx, h_T)^i \rightarrow MLP^i \rightarrow (\widehat{\boldsymbol{y}}_{\boldsymbol{t-H:t}}, \widehat{\boldsymbol{y}}^{\boldsymbol{r}}_{\boldsymbol{t-H:t}}, \widetilde{\boldsymbol{y}}^{\boldsymbol{n}}_{\boldsymbol{t-H:t}}, \boldsymbol{\mathcal{A}})^i \qquad (19)$$

The next step involves selecting the optimal branch among all ν pipelines and reducing it to a linear interpolation of the mixed models. The following steps were taken to achieve this:

1. From the selected ν models, the model that achieves the best (maximum) prediction accuracy is selected.
2. Calculation of the average value of the predicted target variable's residual values.

$\overline{\boldsymbol{y}}^{\boldsymbol{r}}_{\boldsymbol{t-H:t}} = \frac{\mathbf{1}}{\boldsymbol{\nu}} \sum_{\mathbf{i=1}}^{\boldsymbol{\nu}} \widehat{\boldsymbol{y}}^{\boldsymbol{r}}_{\boldsymbol{i,t-H:t}}$.

3. For each target **f** indicator and **q**-the seasonality characteristic, the value $\bar{y}^s_{q,f}$ (e.g., all months of the first quarter) is calculated according to the mean square deviation.

$\hat{y}^*_{q,f} = \gamma \times (\tilde{y}^n_{q,f} + \alpha \times \bar{y}^r_{q,f}) + (1-\gamma) \times \bar{y}^s_{q,f}$,

where $\alpha, \gamma \in [0,1]$ is chosen by selecting the residual prediction made on the seasonally adjusted training data as close as possible to the true values of the target features.

5. The parameters, weights, and estimates of the constructed models are stored for feedback in the calculations of subsequent layers.

The output from each branch of the LSTM-Attention model is fed into the nudging layer, creating a single branch. Let's discuss the functionality of the nudging layer.

4.3 Nudging Layer

Let us represent the Nudging correction process of the nudging model for the univariate multi-step prediction in the form of the following Algorithm 1.

Algorithm 1: Nudging correction for the i-th branch in the univariate multiple-step prediction

1: Input: $\boldsymbol{\Theta_1^i}$- LSTM-Attention network weights, $X^i, y^i -$ last window training data, $\mathcal{A}_1^i$- accurcy prediction from the previous stage, $\mathcal{A}^{tresh}$- threshold of the accuracy for all branches.
2: Outcome: Corrected LSTM-Attention network weights, $\mathcal{A}_k^i$- accuracy;
3: k←1;
4: if $\mathcal{A}_k^i \geq \mathcal{A}^{tresh}$: **return** $\boldsymbol{\Theta_k^i}, \mathcal{A}_k^i$;
5: Initialize univariate LSTM-Attention model $\boldsymbol{M}_k$ with parameters $\boldsymbol{\Theta_k^i}$;
6: Compile $\boldsymbol{M}_k$;
7: $\boldsymbol{\Theta_k^i} \leftarrow$ **Train** $(\boldsymbol{M}_k,\ X^i, y^i)$;
8: $X_T^i \leftarrow$ **LastWindow** (X^i);
9: $\check{y}_e^i \leftarrow$ **Predict** $(\boldsymbol{M}_k, X_e^i)$; # predict for the last window
10: $\check{y}_{e-1}^i \leftarrow$ **GetFromHistory** (X^i, y^i);
11: $\hat{y}_n^i \leftarrow$ **CalculateNudgePrm** $(\check{y}_e^i, \check{y}_{e-1}^i y_e^i, \mathcal{A}_k^i)$;
12: $\boldsymbol{\Theta_k^i} \leftarrow$ **CalibrateModel**$(\boldsymbol{M}_k, \check{y}_e^i, \hat{y}_n^i)$
13: $\mathcal{A}_k^i \leftarrow$ **CalculateAccuracy**$(\hat{y}_e^i)$;
14: k←k+1;
15: goto 4;

Here, we will focus on only two points of the proposed algorithm, points 11 and 13.

4.3.1 Calculation of the nudging parameters

Let's consider the operations in order

1. The nudge correction is calculated according to the selected policies and system parameters

$$\check{y}_{nudged} = \begin{cases} \check{y}_e - \delta, \mathcal{A}_k \geq \mathcal{A}_{tresh} \ \& \ y_e < \check{y}_e \\ \check{y}_e + \delta), \mathcal{A}_k \geq \mathcal{A}_{tresh} \ \& \ y_e > \check{y}_e \end{cases}$$

Where $\delta =$ FEEDBACK_SHIFT - is a system parameter.

2. The seasonality y_{season} component is calculated from historical data.

3. The goal is calculated by weighing the base and seasonal components.

$$\hat{y}_{nudged} = (1-\lambda) \times \check{y}_{nudged} + \lambda \times y_{season}, \quad \lambda \in [0,1].$$

4. The forecast trust region $tr = [c_{min}, c_{max}]$ is formed according to the previous prediction.

5. The goal is clipped according to the confidence interval $\hat{y}_{nudged} = clip(\hat{y}_{nudged}, tr)$.

4.3.2 Calibrate the Model

Calibration is performed in the following steps:

5 The deviation between the predicted and nudged features$\Delta = \check{y}_e^i - \hat{y}_n^i$.

1. **Weights** and **bias** will be taken from the model.
2. Corrected **bias** = **bias** + Δ

Algorithm 1 provides adaptability of the prediction process, since the number of k-steps in the prediction of each feature can be different, and it can be adjusted automatically or manually.

5 Experiment

5.1 Datasets

To evaluate practical forecasting performance, the proposed model was tested on both synthetic data and the following real datasets:

1. Georgia's quarterly macroeconomic dataset (2008–2024) from geostat.ge, containing 18 variables, eight of which were forecast targets.

2. Macroeconomic datasets for Germany, South Korea, and the United States from fred.stlouisfed.org, containing five key indicators: GDP (million $), GDP per capita $, Exports (million $), Imports (million $), and exchange rates (national currency to USD); for the U.S., the dataset also includes Broad Money M3 (million $).

5.2 Hyperparameters

The choice of hyperparameter values for a model greatly impacts its efficiency and accuracy. To optimally select hyperparameters, we used the Keras tuner, which, together with TensorFlow, offers built-in search algorithms and tools for efficiently finding the best hyperparameters within a defined space, thereby enhancing model accuracy and generalization.

Table 1. Hyperparameter values used for training our hybrid model

Parameter	Value
Sequence Length (L)	128
Stride (S)	1
Hidden Dimension (per BiLSTM layer)	64
BiLSTM Layer Count	2
Dropout Probability	0.2
Optimizer	Adam
Learning Rate	0.001
Batch Size	64
Number of Epochs	50
Sliding windows Forecast horizon size	4
Forecast horizon size	4

Source: created by the authors

The table 1 depicts the hyperparameters for the Georgia budget forecasting example. The

FEEDBACK_SHIFT system parameter for the nudging layer has been chosen to 0.15. Most hyperparameters were kept consistent across datasets, except for the number of epochs.

5.3 Experimental design

The first experiment uses quarterly Georgian macroeconomic indicators for 2008Q1-2023Q4 and forecasts 2024Q1-2024Q4. The target variables include GDP, GDP per capita, imports, state revenues, broad money M3, average exchange rate, number of employees, and agricultural production output. The proposed model was executed three times to evaluate stability across repeated runs.

The experiment uses a controlled AI-platform comparison. The same CSV input and the

same forecasting instructions were submitted to ChatGPT 5.4, Gemini 3.1, Grok 4.3, and Claude Sonnet 4.6. The instruction explicitly required the platforms to use only the provided 2008-2023 quarterly Georgian data and not to retrieve external web data. The platform outputs were then compared with the realized 2024 values using the same average-accuracy metric as the proposed model.

The second experiment evaluates external robustness. Germany and South Korea were tested with five quarterly macroeconomic indicators for 2008Q1-2023Q4, while the United States was tested with five quarterly indicators for 1999Q1-2023Q4. In each case, the model generated forecasts for 2024Q1-2024Q4 and was evaluated against realized 2024 values.

Table 2. shows the platform comparison protocol planned for the experiments. Table 3 shows the experiment execution time by platform

Table 2. AI-platform comparison protocol used in the experiment

Component	Protocol
Input data	Quarterly Georgian macroeconomic dataset, 2008Q1-2023Q4.
Forecast target	All available quarterly macroeconomic parameters for 2024Q1-2024Q4.
Instruction constraint	Use only the provided CSV data; do not use external web data.
Requested methods	Time-series methods, deep-learning logic, BiLSTM-Attention reasoning, ETS, ARIMA
Evaluated platforms	ChatGPT 5.4, Gemini 3.1, Grok 4.3, Claude Sonnet 4.6.
Proposed model runs	Three consecutive executions of the BiLSTM-Attention-Nudging model.
Evaluation metric	Average accuracy computed for each indicator and quarter using the same metric.

Source: created by the authors

Table 3. Experiment execution time

System	Execution date and time
Grok 4.3	18.04.2026, 00:01-00:03
Gemini 3.1	18.04.2026, 00:03-00:07
ChatGPT 5.4	18.04.2026, 09:32-09:40
Claude Sonnet 4.6	18.04.2026, 09:40-09:44
Proposed model, run I	18.04.2026, 12:00-12:02
Proposed model, run II	18.04.2026, 12:24-12:25
Proposed model, run III	18.04.2026, 12:37-12:39

Source: created by the authors

5.4 Results

The first experiment's results on Georgian datasets are shown in Figure 3 and Figure 4 (Appendix), and Table 4 (Appendix). Figure 3 illustrates the system's training using Georgia's historical budget data (2008-2023) and a GDP forecast for 2024 (in millions of dollars) based on 2023 test data. Figure 4 (Appendix) compares the 2024 GDP per capita and overall GDP forecasts with actual data. Table 4 (Appendix) demonstrates the model's consistent performance across three runs, with scores of 94.09%, 93.91%, and 94.16%. The best single AI-platform baseline in the same setting was Grok 4.3 with 94.30%, followed by Gemini 3.1 with 93.74%, Claude Sonnet 4.6 with 91.31%, and ChatGPT 5.4 with 90.81%. These results should not be interpreted as a universal ranking of AI systems. They only describe the behavior of the tested platforms under the specific prompt, dataset, and execution times used in this experiment.

The key conclusion is that the proposed model is competitive with large AI platforms while remaining transparent at the methodological level. Unlike the platform baselines, the proposed model has an explicit data pipeline, equation-defined forecast operator, fixed hyperparameters, a defined loss function, and a bounded correction mechanism. This improves reproducibility and makes the results more suitable for academic evaluation.

The second experiment focuses on utilizing external data sets to comprehensively test the system. To accurately assess the system's performance, it is crucial to conduct tests using multiple data sets over different time intervals. In this context, the model was evaluated with data from three additional countries, each featuring unique economic structures and characteristics.

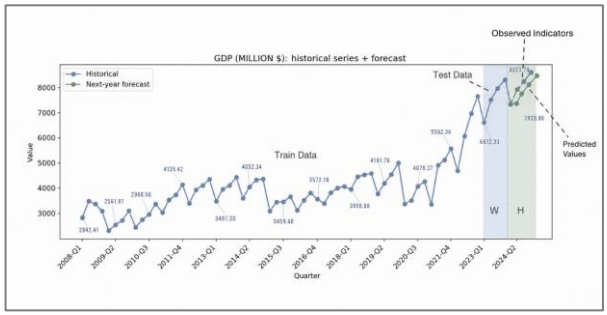


Fig. 3: Forecast of the Georgian budget for 2024 based on training with 2008–2023 historical data

Source: created by the authors.

Germany and South Korea were tested on 2008Q1-2023Q4 quarterly data, while the United States was tested on 1999Q1-2023Q4 quarterly data. In all three cases, the forecast target was 2024Q1-2024Q4.

The model achieved 96.35% average accuracy for Germany, 96.05% for South Korea, and 97.04% for the United States. The strongest country-level result was obtained for the United States, where the longer 1999-2023 history provides a broader training window. The German and South Korean experiments confirm that the model can remain accurate under different units, scales, and macroeconomic profiles.

The assessments are presented in full according to macroeconomic characteristics in Table 5.

Table 5. Cross-country validation by indicator, average 2024 accuracy, %

Indicator group	Germany	South Korea	United States
GDP	98.86	97.23	97.32
GDP per capita	98.94	97.34	97.98
Exports	94.38	91.96	97.43
Imports	92.90	95.34	97.09
Exchange rate	96.65	98.41	95.38
Overall average	96.35	96.05	97.04

Source: created by the authors

6 Conclusion and Future Work

We analyzed the strengths and weaknesses of existing deep neural network models for various tasks, with emphasis on forecasting macroeconomic indicators, and found the LSTM-Attention-Nudge hybrid to be especially effective. Based on this model and our architecture, we have developed a new forecasting system. Experiments on synthetic and real data showed high accuracy and robustness to certain parameter changes, while indicating areas for further improvement.

Next steps:

1. Provide rigorous mathematical justification of the model to strengthen its validity.
2. Use machine learning to optimize the nudge model's parameters.
3. Incorporate semantic-domain structural constraints into the forecasting model to enhance explainability.

Declaration of Generative AI and AI-assisted Technologies in the Writing Process

During the preparation of this work, the author(s) utilized artificial intelligence (AI) technologies solely for the purpose of linguistic refinement and grammatical correction. Following the application of these tools, the author(s) comprehensively reviewed, verified, and edited the resulting text. The author(s) maintain full accountability and responsibility for the accuracy, integrity, and final content of the publication.

References:

[1] Defu Cao, Yixiang Zheng, Parisa Hassanzadeh, Simran Lamba, Xiaomo Liu, and Yan Liu. Large Scale Financial Time Series Forecasting with Multi-faceted Model.

In 4th ACM International Conference on AI in Finance (ICAIF '23), November 27–29, 2023, Brooklyn, NY, USA. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3604237.3626868

[2] Ta-Chung Chi, Ting-Han Fan, Raffaele M. Ghigliazza, Domenico Giannone, Zixuan (Kevin) Wang, Macroeconomic Forecasting and Machine Learning, arXiv:2510.11008v1 [econ.EM] 13 Oct 2025

[3] Cook, Thomas R., and Aaron Smalter Hall. "Macroeconomic Indicator Forecasting with Deep Neural Networks." Federal Reserve Bank of Kansas City, Research Working Paper 17-11, September. Available at External Link, https://doi.org/10.18651/RWP2017-11

[4] Qi Tang, Ruchen Shi, Tongmei Fan, Yidan Ma, and Jingyan Huang, Prediction of Financial Time Series Based on LSTM Using Wavelet Transform and Singular Spectrum Analysis, Hindawi Mathematical Problems in, Engineering, Volume 2021, Article ID 9942410, 13 pages, https://doi.org/10.1155/2021/9942410

[5] Ammara Laeeq, Jie Li, Usman Adeel, A Hybrid LSTM–Attention Model for Multivariate Time Series Imputation: Evaluation on Environmental Datasets, Mach. Learn. Knowl. Extr. 2026, 8, 18, https://doi.org/10.3390/make8010018

[6] Bryan Lima, Sercan O. Ar_kb, Nicolas Loe_b, Tomas P_sterb, Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting, arXiv:1912.09363v3 [stat.ML] 27 Sep 2020

[7] Zurab Bosikashvili, Tamar Bosikashvili, Ketevan Bosikashvili - Using Blocking Meta-heuristics in the Models of Deep Learning, International scientific practical conference modern challenges and achievements in information and communication technologies – 2023, Articles of the conference © Publishing House - Technical University, 2023 ISBN 978-9941-512-06-, http://www.gtu.ge. (Accessed Date: September 15, 2026).

[8] Pierre Beaujuge, Multivariate Time Series Forecasting with a Bidirectional LSTM: Building a Model Geared to Multiple Input Series, Medium, Sep 24, 2020, https://medium.com/@pierre.beaujuge/multivariate-time-series-forecasting-with-a-bidirectional-lstm-building-a-model-geared-to-4f020a160636 (Accessed Date: September 15, 2026).

[9] Gholam Reza Alipour Modab, Attention-driven deep learning models for multivariate time series forecasting of reservoir water levels, Water Sci Technol (2025) 92 (12): 1650–1668. https://doi.org/10.2166/wst.2025.182

[10] Dominik Stempien, and Robert Slepaczuk, Hybrid Models for Financial Forecasting: Combining Econometric, Machine Learning, and Deep Learning Models, https://doi.org/10.48550/arXiv.2505.19617.

[11] Kehang Feng, Yuxin Zhang, LSTM-Attention: A Novel Deep Learning Model for Sustainable Development Goals Achievement Prediction, MLSC '25: Proceedings of the 2025 International Symposium on Machine Learning and Social Computing, Pages 46 – 50, https://doi.org/10.1145/3778450.3778457

[12] Navid Hajizadeh, S. Yazdani, Sara Ershadi-Nasab, Attentional Bi-LSTM for Multivariate Time Series Forecasting on Edge Devices: A Case Study on NanoPi Neo Plus2, Published in International Conference on, 28 October 2025, Computer Science, Environmental Science, doi: 10.1109/ICCKE68588.2025.11273888

[13] Inna Polichtchouk, Simon Lang, Sarah-Jane Lock, Michael Maier-Gerber, Peter Dueben, Hybrid ensemble forecasting combining physics-based and machine-learning predictions through spectral nudging, 5 Mar 2026, https://doi.org/10.48550/arXiv.2603.05570

[14] Jiangcheng Zhu, Shuang Hu, Rossella Arcucci, Chao Xu, Jihong Zhu, and Yi-ke Guo, Model Error Correction in Data Assimilation by Integrating Neural Networks, BIG DATA MINING AND ANALYTICS, ISSN222096-0654 02/05, pp83–91, Volume 2, Number 2, June 2019, doi: 10.26599/BDMA.2018.9020033

[15] Harbir Antil, Rainald Löhner, Randy Price, Data Assimilation with Deep Neural Nets Informed by Nudging, 22 Nov 2021, https://doi.org/10.48550/arXiv.2111.11505

[16] Yingjun Peng, Yulong Bai, Xufeng Wang, Qinghe Yu, Xiaoxin Yue, Nudging-Based Data Assimilation Method for Error Correction Coupled With Huber Loss Functions and BiLSTM-GRU Hybrids, November 2025, Journal of Advances in Modeling Earth Systems, doi: 10.1029/2025MS005306

[17] Hassanzadeh, P., Chattopadhyay, A., Bach, E., Nabizadeh, E. (2023): Deep learning-enhanced ensemble-based data assimilation for high-dimensional nonlinear dynamical systems, XXVIII General Assembly of the International Union of Geodesy and Geophysics (IUGG) (Berlin 2023).https://doi.org/10.57757/IUGG23-3118

[18] Jaemin Oh, Jinsil Lee, Youngjoon Hong, Machine Learning-Based Nonlinear Nudging for Chaotic Dynamical Systems, 7 Aug 2025, https://doi.org/10.48550/arXiv.2508.05778; Available at: https://arxiv.org/abs/2508.05778

[19] Sun, H., Lei, L., Liu, Z., Ning, L., & Tan, Z. M. (2025). An online paleoclimate data assimilation with a deep learning-based network. Journal of Advances in Modeling Earth Systems, 17(6), https://doi.org/10.1029/2024MS004675

[20] Heaney, C. E., Tang, J., Yan, J., Guo, D., Ipock, J., Kaluvakollu, S., et al. (2024). Data assimilation with machine learning for dynamical systems: Modelling indoor ventilation. Physica A: Statistical Mechanics and its Applications, 643, 129783. https://doi.org/10.1016/j.physa.2024.129783

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors contributed equally to the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

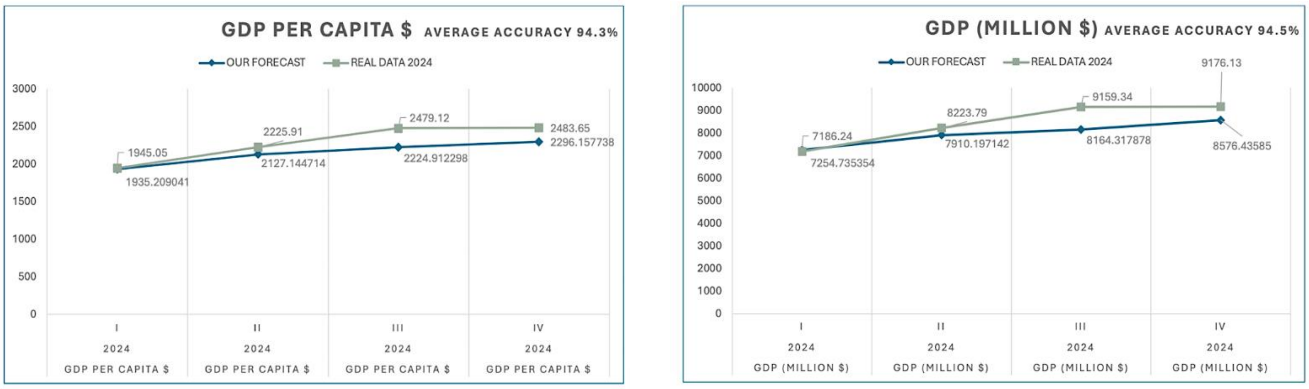
Conflict of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0 https://creativecommons.org/licenses/by/4.0/deed.en_US

APPENDIX



1. b)

Fig. 4: a) Forecast results of the GDP PER CAPITA ($) indicator for the four quarters of 2024 in relation to the actual value; b) Forecast results of the GDP (MILLION $) indicator for the four quarters of 2024 in relation to the actual value

Source: created by the authors.

Table 4. Average 2024 accuracy by Georgian indicator and tested system, %

Indicator	ChatGPT	Gemini	Grok	Claude	Our Model run I	Our Model run II	Our Model run III
GDP (M $)	87.39	94.02	96.94	91.27	94.83	94.28	94.69
GDP PER CAPITA $	89.81	94.60	98.26	92.13	94.49	94.40	94.60
Imports (M $)	86.37	95.58	91.12	89.26	92.57	92.16	92.88
State Revenues (M $)	77.69	94.40	89.98	82.03	92.06	92.36	92.10
Broad Money M3 (M $)	98.70	95.94	98.78	94.81	92.00	92.02	92.00
AV. Exchange USD-$	99.14	99.03	99.08	95.36	97.50	97.11	98.07
Number Of Employees	97.46	99.18	98.23	97.05	97.15	96.31	97.29
Agricult. Products Outp. (M $)	89.94	77.14	82.00	88.61	92.10	92.66	91.65
Overall average	**90.81**	**93.74**	**94.30**	**91.31**	**94.09**	**93.91**	**94.16**

Source: created by the authors