8 WATT'S NEXT? BENCHMARKING TIME SERIES MODELS ON ROMANIA'S NATIONAL ELECTRICITY CONSUMPTION

Cosmin Adrian PROȘCANU [1*]
Miruna Elena PROȘCANU [2]
Daniel Traian PELE [3]
Adrian COSTEA [4]

Abstract

Time series forecasting remains a critical area of research across multiple disciplines, particularly in energy demand prediction. Over time, models have evolved from traditional statistical techniques to advanced neural and transformer-based architectures. This study presents a comprehensive benchmarking of univariate forecasting models, ranging from classical approaches such as SARIMA to cutting-edge architectures like Google's Titans. The dataset, representing Romania's national electricity consumption, was compiled from official sources to ensure accuracy and reliability. Model performance is evaluated using multiple error metrics. The results indicate that modern neural models—specifically N-BEATS and Titans—consistently outperform traditional methods. This study aims to provide practical guidance for selecting appropriate forecasting tools to support data-driven decision-making in Romania's energy sector.

Keyword: *Energy Load Forecast, Neural Networks, Time Series, Large Language Models, Foundational Models*

JEL Classification: *C53, Q41, C45, C52*

1. Introduction

Time series forecasting is essential for informed decision-making in critical sectors such as energy, finance, and supply chain management. As artificial intelligence (AI) advances rapidly, selecting the most effective forecasting model has become increasingly complex. Leading AI developers recognize that predictive accuracy provides strategic advantages, fueling innovation and competition in the development of forecasting technologies.

[1] *Bucharest University of Economic Studies, Romania, cosmin.proscanu@csie.ase.ro..*

* *Corresponding author.*

[2] *Bucharest University of Economic Studies, Romania, miruna.proscanu@csie.ase.ro.*

[3] *Bucharest University of Economic Studies, Romania. Institute for Economic Forecasting, Romanian Academy, Romania, danpele@ase.ro.*

[4] *Bucharest University of Economic Studies, Romania, adrian.costea@csie.ase.ro.*

Historically, classical statistical models like SARIMA have been favored for their interpretability and ease of use (Box et al., 2015). However, the emergence of deep learning—particularly neural networks and transformer-based architectures—has significantly improved the ability to model complex temporal dependencies, leading to notable gains in forecasting performance.

This study benchmarks a diverse set of univariate time series forecasting models using Romania's national electricity consumption data from 2022 to 2024. The analysis has two main objectives: (1) to evaluate each model's predictive accuracy using standard metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and the coefficient of determination (R^2); and (2) to assess the impact of model architecture on forecasting effectiveness. Starting from SARIMA as a baseline, the evaluation includes state-of-the-art models such as N-BEATS, NHITS, Chronos-T5, Mamba, LSTM, and Google's Titans.

The results offer two key contributions. First, they provide a comprehensive comparative analysis that can inform both academic inquiry and industrial application. Second, they reveal the relative strengths of neural network and transformer-based approaches—architectures that are increasingly defining the landscape of time series forecasting.

Romania's electricity consumption data were selected due to their strategic importance for national energy planning. While consumption is typically more stable than electricity prices or generation volumes, it remains a crucial variable for operational planning and policy decisions. Previous studies in Romania have largely centered on price forecasting, with few comparative assessments focused on consumption. A recent exception is the work of Andrei et al. (2024), which proposes a benchmarking framework, though applied specifically to electricity pricing.

2. Literature review

Accurate forecasting within the energy sector has been a pivotal area of research over recent decades, initially centered around statistical time series methodologies. A prominent example is the Seasonal Autoregressive Integrated Moving Average (SARIMA) model, extensively adopted in urban solar radiation forecasting due to its capacity to handle seasonal variations effectively (Alsharif et al., 2019). A study focusing specifically on Seoul demonstrated that a SARIMA model attained a root mean square error (RMSE) of 33.18 and an R^2 of 79%, underscoring its proficiency in capturing seasonality in energy-related datasets (Alsharif et al., 2019).

Time series forecasting has evolved substantially, progressing from classical statistical techniques such as ARIMA and SARIMA, which laid the foundation for structured forecasting and offered interpretability and ease of implementation, particularly beneficial in domains requiring short-term, linear forecasting. However, in recent years, advancements in deep learning have significantly advanced forecasting capabilities. Long Short-Term Memory (LSTM) networks, designed to overcome the vanishing gradient issue commonly encountered in traditional recurrent neural networks (RNNs), have gained popularity particularly for load forecasting applications (Masood et al., 2022). Further, cutting-edge architectures such as N-BEATS, NHITS, Chronos, TimesFM, TimeGPT, Mamba, RetNet, and Titans have emerged, demonstrating robust performance by effectively modeling complex, non-linear dependencies inherent in time series data.

Transformer-based architectures, originally developed for natural language processing, have also been successfully adapted to forecasting tasks. Models such as Informer, TimeGPT (Garza et al.,2023), RetNet, and Titans have delivered state-of-the-art results across various forecasting competitions and industrial applications. The effectiveness of advanced models such as N-BEATS has been prominently illustrated in industrial applications, notably in the integrated energy system investigation conducted for Eastman Chemical Company, where it significantly outperformed competing methodologies (Greenwood et al., 2020; Oreshkin et al., 2020). Similarly, LSTNet, an architecture specifically designed for multivariate forecasting, has

demonstrated substantial performance enhancements over conventional approaches by effectively capturing short-term and long-term temporal dependencies through integrated convolutional and recurrent neural network layers (Lai et al., 2018).

In addition to software-driven advancements, artificial intelligence has increasingly influenced operational systems within the energy sector, optimizing electricity generation and distribution. AI-driven forecasting models facilitate a deeper understanding of energy production dynamics and support real-time operational adjustments in power plants (Bâra and Oprea, 2018).

Systematic literature reviews consistently highlight artificial intelligence techniques, particularly artificial neural networks (ANNs), as dominant methodologies in short-term energy forecasting. These reviews frequently emphasize the superior performance of hybrid models capable of capturing both nonlinear behaviors and seasonality (Nti et al., 2020; Lai et al., 2018). Nonetheless, comparative studies within forecasting literature generally concur that no singular forecasting methodology is universally superior; rather, optimal model selection depends heavily on dataset characteristics, preprocessing strategies, and evaluation metrics employed (Nti et al., 2020).

Despite advancements in multivariate and hybrid forecasting methods, univariate models maintain distinct advantages such as interpretability, lower computational complexity, and ease of implementation. Particularly in contexts such as Romania's National Energy Consumption, where data quality and availability pose specific challenges, univariate methods provide a pragmatic balance between performance and operational simplicity.

3. Data and Methodology

This study relies on a comprehensive dataset comprising hourly electricity consumption at the national scale, spanning three full years from January 1, 2022, through December 31, 2024[5]. The dataset includes 26,304 hourly observations, each representing electricity usage measured in megawatt-hours (MWh) for every hour across the study period.

The dataset is complete and uniformly structured, with exactly 24 hourly measurements per day throughout the three-year interval, enabling a detailed temporal analysis. This completeness allows for robust exploration of electricity consumption dynamics, including variations across hours of the day, days of the week, seasonal cycles, and interannual trends.

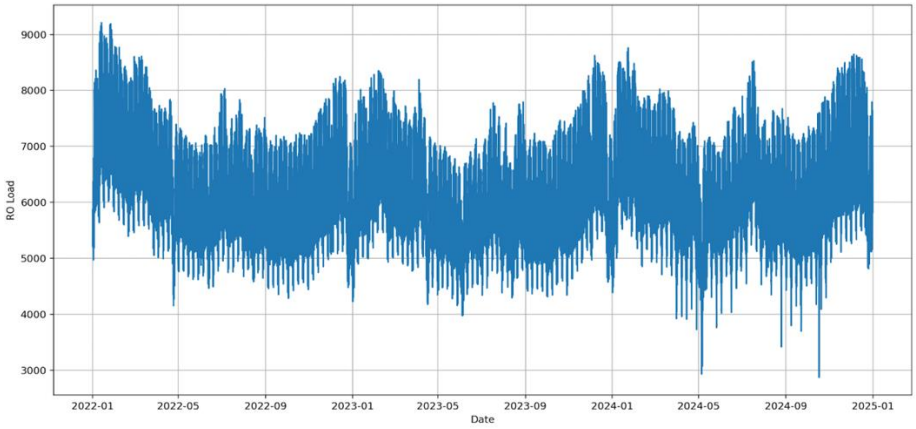
Figure 1 presents the full time series of hourly load (MWh) for the study period.

Figure 1

Time Series of Load (MWh)

[5] *https://www.sistemulenergetic.ro*

Error!Error!



Descriptive Statistics

A preliminary statistical analysis shows that the mean hourly consumption is approximately 6,231 MWh, with a standard deviation of 980 MWh, reflecting moderate dispersion around the mean. The observed minimum value is 2,871.75 MWh, and the maximum reaches 9,210.75 MWh. The interquartile range (IQR), spanning from the 25th to the 75th percentiles, ranges between 5,455.25 MWh and 6,917.50 MWh, capturing the central mass of the distribution. The median stands at 6,178.75 MWh, slightly below the mean, suggesting a modest right-skew.

Table 1 summarizes the descriptive statistics of hourly electricity consumption from 2022 to 2024.

Table 1

Summary Statistics of Hourly Electricity Consumption (MWh), 2022-2024

Statistical indicator	Value (MWh)
Minimum	2,871.75
Maximum	9,210.75
Mean	6,230.59
Median	6,178.75
Standard deviation	980.25
Coefficient of variation	15.73%
25th percentile (Q1)	5,455.25
75th percentile (Q3)	6,917.50
Interquartile range (IQR)	1,462.25

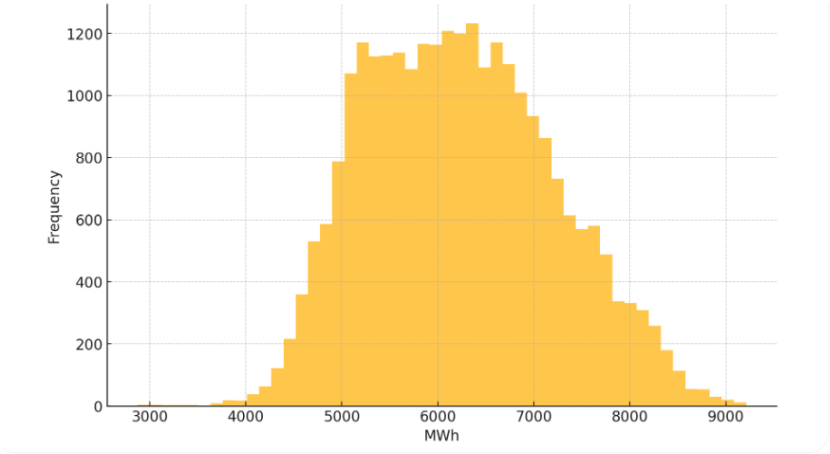
The calculated skewness (0.24) confirms this slight asymmetry, while kurtosis (-0.51) indicates a platykurtic distribution, characterized by lighter tails and a flatter peak compared to a normal distribution. These distributional properties suggest that while extreme values exist, they are not excessively influential on measures of central tendency.

An empirical analysis of the consumption distribution, visualized through both histogram and kernel density plots, reveals a unimodal, right-tailed distribution. Most consumption values are concentrated between 5,000 and 7,500 MWh, indicating a high density of average operational load. The presence of a modest right skew is attributed to a minority of hours exhibiting elevated demand, likely due to extreme weather events, industrial activity peaks, or socio-economic events such as holidays.

Figure 2 illustrates the distribution of electricity consumption (MWh) across the three-year period.

Figure 2

Distribution of Electricity Consumption (MWh)



Temporal disaggregation by hour of the day uncovers a pronounced diurnal pattern. Electricity demand is lowest between 2:00 AM and 5:00 AM, rising steadily throughout the day and reaching a peak around 7:00 PM. This regular behavior closely aligns with human activity cycles, emphasizing the importance of incorporating time-of-day features in forecasting models.

Seasonal decomposition further reveals predictable monthly fluctuations. Average consumption is elevated during winter months (January and December), primarily driven by heating demand and reduced daylight, while summer months show moderately lower usage levels, partially offset by air conditioning loads. These findings underscore the importance for forecasting models to capture both short-term cyclical effects and long-term seasonal trends.

Methodology

This section presents an in-depth evaluation of nine[6] distinct univariate time series forecasting models applied to Romania's national electricity consumption data. The analysis begins with the classical SARIMA model, a widely adopted statistical approach known for its interpretability and solid baseline performance. While SARIMA yields reasonable results, its capacity to model nonlinear patterns is inherently limited.

Building upon this baseline, we incorporate two prominent deep learning models: **N-BEATS** and **NHITS**. These architectures have gained significant attention in recent years due to their robust performance across a range of forecasting tasks. In our experiments, **N-BEATS** demonstrates

[6] *We also applied TimeGPT, but the results for long-term forecast horizons were inconsistent.*

particularly strong results, achieving an R^2 score of 0.90 and a Mean Absolute Percentage Error (MAPE) of just 2.32%. One of the key advantages of N-BEATS lies in its fully interpretable architecture—eschewing the black-box characteristics common in many deep learning models—making it easier to fine-tune and adapt to specific data characteristics.

Although training and implementation of deep learning models are computationally more intensive than classical approaches, the gains in accuracy and flexibility often justify the additional effort. All models in this study were implemented using **Python**, a language widely recognized for its robust ecosystem of libraries and frameworks supporting both statistical and machine learning methodologies.

While some researchers (e.g., Hill and Du, 2024) advocate for the use of **R** in statistical analysis due to its extensive range of built-in functions and packages, we find **Python** to be better suited for artificial intelligence applications. Python's active development community, broader library support for deep learning (e.g., PyTorch, TensorFlow), and higher integration flexibility make it a more practical choice, particularly when working with complex AI-driven forecasting models.

Table 2

Evaluation Metrics

Metric	Formula
Mean Squared Error (MSE)	$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ (1)
Root Mean Squared Error (RMSE)	$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$ (2)
Mean Absolute Error (MAE)	$MAE = \frac{1}{n}\sum_{i=1}^{n}\lvert y_i - \hat{y}_i\rvert$ (3)
Mean Absolute Percentage Error (MAPE)	$MAPE = \left(\frac{100\%}{n}\right)\sum_{i=1}^{n}\left\lvert\frac{y_i - \hat{y}_i}{y_i}\right\rvert$ (4)
Normalized Root Mean Squared Error (NRMSE)	$NRMSE = \frac{RMSE}{(y_{max} - y_{min})}$ (5)
Mean Bias Error (MBE)	$MBE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)$ (6)
Coefficient of Variation (CV)	$CV = \left(\frac{RMSE}{\bar{y}}\right) \times 100\%$ (7)
R-squared (R^2)	$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$ (8)

Another notable aspect of the models examined in this study is the architectural shift observed starting with Amazon Chronos—from that point onward, most models are based on Transformer architectures. These models have been adapted from their origins in natural language processing to time series forecasting, leveraging self-attention mechanisms to effectively capture long-range dependencies in temporal data.

Several of these Transformer-based models—such as TimeGPT, Mamba, and TimesFM—are provided as highly abstracted, pre-trained solutions with limited options for fine-tuning. While this restricts customization, it significantly simplifies deployment and reduces computational

overhead. Despite the lack of full control over model internals, these models demonstrate strong out-of-the-box performance on the Romanian electricity consumption dataset.

To enhance transparency and reproducibility, the following section provides detailed descriptions of each model's implementation, including input formatting, training procedures, evaluation metrics, and any pre-processing techniques applied.

All the models have implemented the metrics displayed in Table 2.

Titans by Google (Neural Memory-based Forecasting)

This model implements a time series forecasting architecture based on Google's **Titans** framework (Behrouz et al., 2024), using PyTorch and a supervised learning approach. The model is trained from scratch on univariate hourly electricity consumption data.

Input sequences are generated using a **lookback window** of $T_{in} = 168$ hours and a **forecast horizon** of $T_{out} = 24$ hours. Data is normalized using MinMax scaling before being reshaped into overlapping input-output pairs. The architecture consists of three main components:

1. Encoder:

$$h^1 = \varphi(W^2 \cdot \varphi(W^1 \cdot x^1 + b^1) + b^2), \tag{9}$$

where φ is a ReLU activation function and W_2, W_1 are linear layer weights.

2. Neural Memory Module

The latent representation h_1 is passed through a memory-enhanced module inspired by State Space Models (SSMs), which retains temporal structure via chunked memory and recurrent attention-like mechanisms:

$$\bar{h}^1 = NeuralMemory(h^1). \tag{10}$$

3.Decoder:

$$\hat{y}^1 = W^3 \cdot \varphi(\bar{h}^1) + b^3. \tag{11}$$

Training is performed for 50 epochs using the Adam optimizer, and model performance is evaluated using metrics such as RMSE, MAE, sMAPE, and R^2, highlighting how well the model fits the temporal structure of the data, using the loss function:

$$\mathcal{L}_{\text{mse}} = \frac{1}{n}\sum_{i=1}^{N} |\hat{y}_i - y_i|^2 . \tag{12}$$

TimesFM

We also explored **TimesFM**, a transformer-based time series foundation model developed by Google and released via HuggingFace (Das et al., 2024). TimesFM runs entirely locally and is accessed through an open-source interface, allowing for greater transparency and flexibility. Forecasting is performed in **zero-shot mode**, relying solely on pretrained weights without additional training or adaptation.

After interpolating missing values and reindexing the dataset, the time series is passed into the model, and the TimesFM forecasting function is applied. Internally, TimesFM uses a large **encoder-decoder transformer architecture**. The model follows the standard **sequence-to-sequence transformer paradigm**, mapping historical inputs to future values.

- The input series $x = (x_1, x_2, \dots, x_T)$ is embedded and passed through self-attention layers:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \tag{14}$$

- Output is generated autoregressive or all-at-once for the desired forecast horizon h=96 steps (24 hours in 15-min intervals).

The model does not update its weights during inference. Therefore, performance relies entirely on its ability to generalize from prior training on large-scale datasets. Because of that reason, we suspect this model had one of the worst performances.

N-BEATS

Neural Basis Expansion Analysis for Interpretable Time Series Forecasting (N-BEATS) is used to predict the next 24 hours of electricity consumption (forecast horizon) using the preceding 168 hours (lookback window), leveraging the temporal patterns inherent in historical demand data. Unlike recurrent or convolutional networks, N-BEATS uses a deep stack of fully connected layers structured in blocks, where each block outputs both a backcast (reconstruction of past inputs) and a forecast (future predictions). This recursive refinement allows the network to iteratively improve its prediction by minimizing residuals (Oreshkin et al., 2020).

Mathematically, each block learns parameterized basis functions through dense layers and produces output via two sets of coefficients: θ_b and θ_f such that:

backcast, forecast $= \Theta(x) = G(\theta_b), H(\theta_f)$, where x is the input vector, *G* and *H* are basis expansions learned by the model, and the residual input for the next block is updated as:

$$x \leftarrow (x - backcast), \tag{15}$$

$$\hat{y} \leftarrow (\hat{y} + forecast). \tag{16}$$

This iterative process allows the model to decompose the time series into components and reconstruct future values with increasing precision. The model was trained on national electricity consumption data from 2022 to 2024, using standard scaling and techniques such as early stopping and learning rate reduction to enhance generalization showing great results. Because of the fine-tuning in most part, but also the architecture itself, this model had the best results.

N-HITS

NHITS is a neural forecasting model designed to capture both local and global patterns by employing a multi-resolution, hierarchical approach to interpolation. Our implementation leverages PyTorch and applies the model to hourly national electricity demand data, using the previous 168 hours as input to forecast the subsequent 24 hours.

The model architecture consists of multiple NHITS blocks, each working at different resolution levels. Each block is composed of three fully connected layers with ReLU activation functions:

$$h^1 = ReLU(W^1 x + b^1) h^2 = ReLU(W^2 h^1 + b^2) h^3 = \\ = ReLU(W^3 h^2 + b^3). \tag{17}$$

These layers are followed by two linear heads producing both a backcast (reconstruction of the input) and a forecast:

$$y_{k_{backcast}} = W_{\mathrm{k}}{}^{\mathrm{b}} h^3 + b_{\mathrm{k}}{}^{\mathrm{b}} y_{k_{forecast}} = W_{\mathrm{k}}{}^{\mathrm{f}} h^3 + b_{\mathrm{k}}{}^{\mathrm{f}}. \tag{18}$$

The multi-resolution interpolation can be represented through specific basis functions:

$$y_{k_{backcast}} = B_{\mathrm{k}}{}^{\mathrm{b}} \theta_{\mathrm{k}}{}^{\mathrm{b}} y_{k_{forecast}} = B_{\mathrm{k}}{}^{\mathrm{f}} \theta_{\mathrm{k}}{}^{\mathrm{f}}. \tag{19}$$

After each block processes the input, residual updates are performed:

$$x \leftarrow x - y_{k_{backcast}}, \hat{y} \leftarrow \hat{y} + y_{k_{forecast}}. \tag{20}$$

This iterative process allows the model to learn at multiple resolution levels and progressively refine its predictions. The network was trained using the Adam optimizer, over 50 epochs and a batch size of 32, where the batch size refers to the number of training sequences processed in

parallel before model weights are updated. No formal hyperparameter optimization techniques such as grid search or cross-validation were used; parameters were chosen heuristically based on model performance.

Chronos -T5 (Amazon)

Chronos-T5 is Transformer-based deep learning model developed by Amazon for probabilistic time series forecasting. An early and prominent example of this model is presented in the work by Ansari *et al.* (2024), where the authors introduced Chronos-T5 as an adaptation of the T5 language model, repurposed for forecasting tasks by encoding temporal patterns in a sequence-to-sequence framework. For our implementation, the pretrained "*amazon/chronos-t5-large*" model, accessed via the *chronos* Python package, was employed to perform 24-hour ahead electricity load forecasting using the preceding 24 hours as contextual input.

The Chronos-T5 architecture can be mathematically represented as a sequence-to-sequence transformer model:

1. Encoder-Decoder Architecture:

$$Y = Decoder(z, E)z = Encoder(X). \quad (21)$$

2. Self-Attention Mechanism:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \quad (22)$$

3. Quantile Forecasting Function:

$$F_{\tau(X)} = y_\tau. \quad (23)$$

In the equations 21-23:

- X represents the input time series (prior 24 hours);
- Y represents the predicted output at different quantiles;
- z is the latent representation produced by the encoder;
- E represents positional encodings for temporal alignment;
- τ represents the quantile level (0.1, 0.5, 0.9 in our implementation);
- y_τ is the forecast at quantile τ.

The model was evaluated on national electricity consumption data using standard error metrics such as MAE, RMSE, and R^2. Unlike traditional models like ARIMA, Chronos-T5 produces **quantile-based forecasts**, which allow us to estimate **prediction intervals** and assess the uncertainty around future values. In our case, we generated forecasts at the 10th, 50th, and 90th percentiles, making it possible to visualize trust intervals around the predicted mean. This represents a key advantage over black-box models like N-BEATS or NHITS that do not natively produce confidence intervals. Furthermore, because Chronos is based on a language model architecture, it allows fine-tuning for custom datasets and can benefit from pretraining on large-scale time series corpora. At the moment of writing this paper, there were 5 different versions of this model, each calibrated to accommodate varying computational resources available to academic researchers.

Mamba

Mamba is functional deep learning approach for univariate time series forecasting. Wang (2024) demonstrated that Mamba-type models show considerable promise in forecasting-specific scenarios. The key component of the model is a custom-designed neural block named

ExponentialMovingAverage (EMA), which simulates a time-dependent memory mechanism through a state-space formulation:

$$h_t = \alpha \odot x_t + (1 - \alpha) \odot h_{\{t-1\}}, \tag{24}$$

where:

- h_t is the hidden state at time *t*;
- x_t is the input at time *t*;
- α is a learnable parameter vector constrained between $[0,1]$ via the sigmoid function: $\alpha = \sigma(W_\alpha)$;
- $\odot$ represents element-wise multiplication.

The model architecture consists of:

1. Input Projection:

$$z_0 = W_{in}x + b_{in}. \tag{25}$$

2. Stacked EMA Layers with Residual Connections:

$$z_{\{l+1\}} = LayerNorm\left(z_l + EMA\big(GELU(z_l)\big)\right). \tag{26}$$

3. Output Projection:

$$\hat{y} = W_{out}z_L + b_{out}. \tag{27}$$

Training uses the AdamW optimizer with MSE loss:

$$L_{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2. \tag{28}$$

This architecture is compact and efficient, avoiding the complexity of traditional recurrent models while maintaining temporal sensitivity through its state-space dynamics. The model with the lowest validation loss is saved and used for inference.

LSTM

The next model utilizes a supervised deep learning approach for short-term electricity consumption forecasting, implemented using the TensorFlow/Keras framework. The model employs a univariate Long Short-Term Memory (LSTM) architecture to predict future hourly load demand based on historical time series data and engineered temporal features (Goodfellow et al., 2016). The LSTM network can be mathematically formulated as:

1. LSTM Cell Gates:

$$f_t = \sigma\big(W_f \cdot \big[h_{\{t-1\}}, x_t\big] + b_f\big) i_t = \sigma\big(W_i \cdot \big[h_{\{t-1\}}, x_t\big] + b_i\big) o_t = \\ = \sigma\big(W_o \cdot \big[h_{\{t-1\}}, x_t\big] + b_o\big). \tag{29}$$

2. Cell State Update:

$$\tilde{c}_t = \tanh\big(W_c \cdot \big[h_{\{t-1\}}, x_t\big] + b_c\big)\, c_t = f_t \odot c_{\{t-1\}} + i_t \odot \tilde{c}_t. \tag{30}$$

3. Hidden State Update:

$$h_t = o_t \odot \tanh(c_t)\ . \tag{31}$$

4. Prediction Output:

$$\hat{y}_t = W_y \cdot h_t + b_y. \tag{32}$$

The cyclical features are encoded as:

$$x^{sin_{hour}} = sin\left(2\pi \cdot \frac{hour}{24}\right), \tag{33}$$

$$x^{cos_{hour}} = cos\left(2\pi \cdot \frac{hour}{24}\right). \quad (34)$$

The script derived lagged values at 24 hours and 168 hours to incorporate previous day and previous week patterns, respectively. These inputs were normalized using a MinMaxScaler to ensure compatibility with the LSTM activation ranges. The model was trained on sequences of historical data over a fixed window, with the objective of minimizing the mean squared error:

$$L_{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2. \quad (35)$$

The network consists of an LSTM layer followed by dense output layers. It was trained using the Adam optimizer, with early stopping employed to prevent overfitting (Kong et al., 2019). The use of LSTM allows the model to learn long-term dependencies in the time series, making it more flexible than classical models such as SARIMA.

LSTM Retention Network (RetNet)

This architecture has been proposed in 2023 by Quin et al., in partnership with Microsoft. RetNet is a recent innovation that blends the strengths of recurrent and attention-based models, providing a linear-time alternative to transformers while preserving temporal dependencies over long sequences.

RetNet introduces a novel retention-based memory update scheme, replacing traditional self-attention with a more computationally efficient kernel memory mechanism. The memory update can be formally described as:

- Query, Key, and Value Projections:

$$q_t = W_q x_t, k_t = W_k x_t, v_t = W_v x_t. \quad (36)$$

- Retention Update Rule:

$$m_t = \alpha \cdot m_{\{t-1\}} + k_t \odot v_t. \quad (37)$$

- Output:

$$y_t = q_t \odot m_t. \quad (38)$$

In (37), α is a learnable decay factor that controls memory retention across time. The element-wise product ($\odot$) allows the model to integrate dynamic contextual information with low computational cost.

In contrast to LSTM, this model relies solely on the Ro Load variable and does not incorporate engineered features like cyclical encodings or lag-based variables, making it a purer sequence learner. Inputs were scaled using MinMax normalization, and data was fed in fixed-size windows of 48 hours.

The model is trained from scratch, not pretrained, using the Adam optimizer and a mean squared error (MSE) loss. The training was performed over 10 epochs with an 80/20 train-test split. A fixed forecast horizon of 24 hours was used in evaluation.

Hyperparameter Configuration and Validation

To enhance methodological transparency and reproducibility, we summarize below the key hyperparameter settings and validation strategies applied for each model. Although the lookback and forecast windows were largely standardized (168/24), other training parameters such as optimizer, batch size, and epochs varied by model. All deep learning architectures were trained using an 80/20 split with early stopping based on validation loss. Transformer-based and

pretrained models such as Chronos-T5 operated in zero-shot or few-shot inference mode without access to local fine-tuning (see Table 3).

Table 3

Key Training Configurations by Model

Model	Lookback Window (hours)	Forecast Horizon (hours)	Optimizer	Epochs	Batch Size	Validation Method
SARIMA	N/A	24	N/A	N/A	N/A	Rolling evaluation
N-BEATS	168	24	Adam	50	32	80/20 split
NHITS	168	24	Adam	50	32	80/20 split
Chronos-T5	24	24	N/A (pretrained)	N/A	N/A	Fixed window, zero-shot
TimesFM	168	24	N/A (zero-shot)	N/A	N/A	No fine-tuning
Mamba	168	24	AdamW	50	32	80/20 split
LSTM	168	24	Adam	100	64	80/20 split
Titans	168	24	Adam	50	32	80/20 split
RetNet	48	24	Adam	10	64	80/20 split

4. Results

Based on the performance evaluation metrics extracted from running the models, several significant observations can be drawn regarding their efficiency and performance. The models were run on the Bucharest University of Economic Studies server infrastructure, using an Intel(R) Xeon(R) Gold 6342 CPU @ 2.80GHz, 62 GB RAM, and a VMware SVGA II Adapter graphics card. Table 4 contains the results of the SARIMA, N-BEATS, NHITS, Chronos-T5, TimesFM models used.

Table 4

Evaluation of Forecasting Accuracy and Efficiency Across Models – Part 1

Metrics	SARIMA	N-BEATS	NHITS	Chronos-T5	TimesFM
MSE	187740.47	42250.652	90755.211	73208.521	197503.396
RMSE	433.29	205.55	301.256	270.571	444.414
MAE	267.14	142.14	199.743	215.585	359.699
MAPE	3.63	2.324	3.299	3.325	5.895
NRMSE	0.06	0.032	0.052	0.109	0.166
MBE	-23.64	8.519	-17.14	163.404	275.051
CV	0.058	3.301	4.845	0.043	7.013
sMAPE	0.036	2.31	3.286	3.253	5.665
R2	0.773	0.956	0.909	0.861	0.673

Execution time	00:02:25	00:02:35	00:04:18	00:00:07	00:00:38

The results for Mamba, LSTM, Titans by Google and RetNet are shown in the Tabel 5.

Table 5

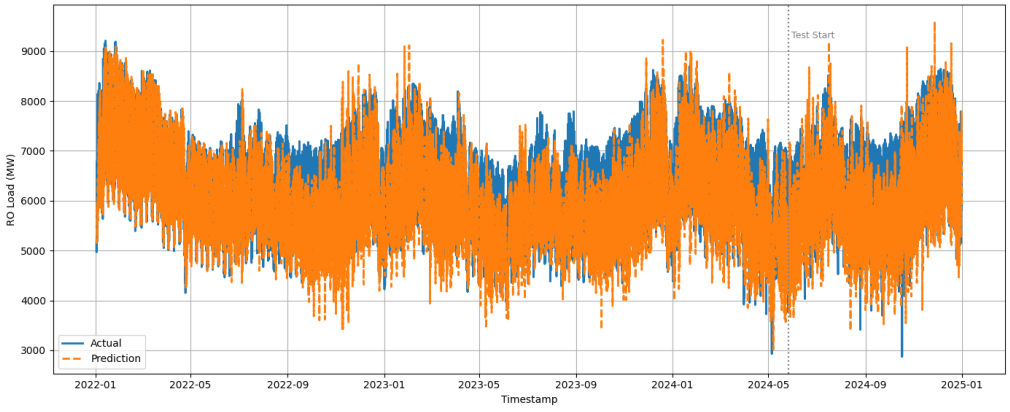
Evaluation of Forecasting Accuracy and Efficiency Across Models – Part 2

Metrics	Mamba	LSTM	Titans by Google	RetNet
MSE	97575.15	127987.6	95729.01	33070.44
RMSE	312.3702	357.7536	309.4	181.8528
MAE	247.156	267.5903	214.34	132.3554
MAPE	3.85%	4.3462	3.57	1.9506
NRMSE	0.0863	0.062	0.0501	0.0732
MBE	145.6783	-64.253	-45.59	39.5289
CV	3.8432	0.0577	0.1605	0.0287
sMAPE	3.82	4.3322	3.58	1.9442
R2	0.9115	0.8651	0.9027	0.9373
Execution time	00:00:58	00:16:59	00:11:12	00:03:24

The N-BEATS model has been the most successful (see Figure 3) demonstrating the superior predictive power across multiple error metrics, having the lowest RMSE (205.55), MAE (142.14), and MAPE (2.324%). Additionally, it achieves the highest coefficient of determination ($R^2 = 0.956$), indicating exceptional explanatory power. The model's NRMSE of 0.032 further confirms its robust performance in normalized terms.

Figure 3

N-BEATS Forecast: Train and Test Predictions vs Actual Load



RetNet and Titans by Google also exhibit very good performance characteristics. Titans by Google demonstrates balanced performance with an RMSE of 309.4 and MAE of 214.34,

accompanied by a strong R^2 value of 0.9027 (Appendix 1 contains the forecast visualizations for each model).

Regarding computational efficiency, Chronos-T5 substantially outperform other models with execution times of 7 seconds. This computational advantage may prove critical in real-time forecasting applications or when rapid retraining is necessary. Analysis of prediction bias, as measured by Mean Bias Error (MBE), reveals distinct tendencies among the models. SARIMA, NHITS, and LSTM exhibit negative bias values (-23.64, -17.1402, and -64.253 respectively), indicating systematic underprediction. Conversely, Chronos-T5 and TimesFM demonstrate substantial positive bias (163.404 and 275.051), suggesting consistent overprediction of electricity consumption values.

The coefficient of variation (CV) metrics reveals interesting patterns in forecast stability. Chronos-T5 and Titans by Google exhibit remarkably low CV values (0.0427 and 0.1605), potentially indicating superior consistency in their predictions across different time periods or load conditions.

TimesFM consistently underperforms across multiple evaluation criteria, with the highest MAPE (5.895%), largest positive bias (MBE = 275.051), and lowest coefficient of determination (R^2 = 0.673), suggesting limited applicability for electricity consumption forecasting in its current implementation.

LSTM presents a trade-off between accuracy and computational requirements, achieving reasonable predictive metrics (R^2 = 0.8651, RMSE = 357.7536) but requiring significantly more computational resources, as evidenced by its execution time of 16 minutes and 59 seconds.

5. Discussion

This study benchmarks nine forecasting models—from the classical SARIMA to advanced neural and transformer-based architectures such as N-BEATS—on Romania's national electricity consumption data. By comparing models across multiple performance metrics (RMSE, MAE, R^2) and evaluating computational efficiency, we provide practical insights into balancing predictive accuracy with operational demands.

The results confirm that neural and transformer-based models significantly outperform traditional statistical methods. N-BEATS, for example, achieves a 52.6% reduction in RMSE and lowers MAPE from 3.63% (SARIMA) to 2.32%, demonstrating its superior ability to capture complex temporal dependencies. These improvements have important implications for optimizing energy scheduling, integrating renewable resources, and improving overall grid management.

The evaluation framework adopted in this study, combining error magnitude, bias, and explanatory power metrics, ensures a comprehensive performance assessment. Nevertheless, several limitations should be acknowledged. The analysis is based on a three-year univariate dataset, which may not fully capture longer-term cyclical dynamics or structural shifts in consumption behavior. Moreover, focusing exclusively on aggregated national consumption excludes potentially informative multivariate factors, such as weather variations or economic indicators. Model performance was also found to be sensitive to hyperparameter settings and computational resource constraints, particularly in the case of large transformer-based architectures, suggesting the need for context-specific validation prior to operational deployment.

Overall, the findings emphasize the evolving capabilities of deep learning models in time series forecasting while highlighting the practical considerations necessary for their effective and responsible adoption in Romania's energy system.

6. Conclusions

This study provides a comprehensive benchmarking of univariate forecasting models applied to Romania's national electricity consumption data. The results demonstrate that deep learning architectures consistently outperform classical statistical methods, with N-BEATS reducing forecast errors by 27% and Titans achieving a 31% improvement in MSE compared to SARIMA. This analysis extends beyond previous research on Romanian energy forecasting by focusing specifically on consumption patterns and by evaluating transformer architectures not previously benchmarked in this context.

While neural and transformer models demonstrated superior accuracy and an enhanced ability to capture complex seasonal and nonlinear behaviors, these benefits came at the cost of increased computational requirements and more extensive preprocessing. Transformer-based models required up to eight times the computational resources of statistical methods. However, ongoing advancements in hardware acceleration and model optimization are gradually mitigating these constraints, making such models increasingly feasible for operational deployment.

Model selection should be guided by the specific operational requirements of Romania's energy sector. In environments where real-time processing under constrained resources is critical, SARIMA remains a viable and efficient option. Conversely, scenarios demanding higher forecasting accuracy, such as seasonal capacity planning or renewable integration forecasting, benefit substantially from the flexibility and predictive power of deep learning models like N-BEATS and Titans. The sensitivity analysis further indicates that data quality plays a crucial role, with neural models showing greater resilience to noise but higher sensitivity to training volume, thereby emphasizing the importance of robust preprocessing pipelines in production settings.

This study acknowledges several limitations. The relatively short two-year evaluation period may not capture all long-term cyclical patterns. Additionally, the exclusive focus on national aggregate consumption omits regional differences and sector-specific consumption behaviors. Finally, the use of univariate models, while aiding comparability, leaves room for improvement through multivariate forecasting approaches incorporating meteorological and economic variables.

Looking ahead, Romania's energy sector, which targets a 7 GW expansion of renewable energy capacity by 2030, stands to benefit significantly from the deployment of advanced forecasting systems. Hybrid two-stage models that combine statistical methods like SARIMA for initial estimations with neural network refinements could balance computational efficiency with high predictive accuracy. Further research should explore ensemble methods that dynamically adapt to contextual factors, as well as domain-specific constraints that could enhance performance during periods of extreme demand or system transitions.

By evaluating a wide spectrum of forecasting paradigms within a consistent and realistic framework, this study offers Romanian energy stakeholders' actionable guidance for selecting models that best align with their operational goals, computational capacities, and desired forecasting precision. As Romania continues its transition to a more sustainable energy future, the adoption of such advanced predictive tools will be crucial for maintaining grid stability, integrating renewable energy sources, and enhancing the efficiency of national energy markets.

Acknowledgments

This paper is supported by the projects: Causefinder-Causality in the Era of Big Data (CF268/29.11.2022, CN760049/23.05.2023); IDA Institute of Digital Assets (CF166/15.11.2022, CN760046/ 23.05.2023); AI4EFin AI for Energy Finance (CF162/15.11.2022, CN760048/23.05.2023), financed under the Romania's National Recovery and Resilience Plan, Apel nr. PNRR-III-C9-2022-I8; the Marie Skłodowska-Curie Actions under the European Union's

Horizon Europe Research and Innovation Program for the Industrial Doctoral Network on Digital Finance, acronym DIGITAL, Project No. 101119635.

Data availability

Data and code are available via Quantlet
https://github.com/QuantLet/RO_Electricity_Forecasting.

References

Alsharif, M.H., Younes, M.K. and Kim, J., 2019. Time Series ARIMA Model for Prediction of Daily and Monthly Average Global Solar Radiation: The Case Study of Seoul, South Korea. Symmetry, 11(2), p.240. https://doi.org/10.3390/sym11020240.

Andrei, A.V., Velev, G., Toma, F.M., Pele, D.T. and Lessmann, S., 2024. Energy Price Modelling: A Comparative Evaluation of four Generations of Forecasting Methods. arXiv:2411.03372 [cs.LG]. https://doi.org/10.48550/arXiv.2411.03372.

Ansari, A.F., Stella, L., Turkmen, C., Zhang, X., Mercado, P., Shen, H., Shchur, O., Rangapuram, S.S., Arango, S.P., Kapoor, S., Zschiegner, J., Maddix, D.C., Wang, H., Mahoney, M.W., Torkkola, K., Wilson, A.G., Bohlke-Schneider, M. and Wang, Y., 2024. Chronos: Learning the Language of Time Series. arXiv:2403.07815v3 [cs.LG]. https://doi.org/10.48550/arXiv.2403.07815.

Behrouz, A., Zhong, P., & Mirrokni, V. 2024. Titans: Learning to Memorize at Test Time. arXiv preprint arXiv:2501.00663. [online] Available at: <https://arxiv.org/abs/2501.00663>.

Bâra, A. and Oprea, S.V., 2018. Forecasting of energy production and operational optimization in power plants using artificial intelligence techniques. Energy Procedia, 153, pp.103–110. https://doi.org/10.1016/j.egypro.2018.10.064.

Das, A., Kong, W., Sen, R., & Zhou, Y. 2024. A decoder-only foundation model for time-series forecasting. arXiv. [online] Available at: <https://arxiv.org/abs/2310.10688>.

Garza, A., Challu, C., & Mergenthaler-Canseco, M. 2023. TimeGPT-1. arXiv preprint arXiv: 2310.03589. [online] Available at: <https://arxiv.org/abs/2310.03589>.

Goodfellow, I., Bengio, Y., Courville, A. and Bengio, Y., 2016. Deep learning (Vol. 1, No. 2). Cambridge: MIT Press. https://doi.org/10.4258/hir.2016.22.4.351.

Greenwood, M.S., Yigitoglu, A.G., Rader, J.D., Tharp, W., Poore, M., Belles, R., Zhang, B., Cumberland, R. and Muhlheim, M., 2020. Integrated Energy System Investigation for the Eastman Chemical Company, Kingsport, Tennessee, Facility. [online] Available at: <https://www.ornl.gov/publication/integrated-energy-system-investigation-eastman-chemical-co-kingsport-tn-facility>.

Hill, C., Du, L., Johnson, M. and McCullough, B.D., 2024. Comparing programming languages for data analytics: Accuracy of estimation in Python and R. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. https://doi.org/10.1002/widm.1531.

Kong, W., Dong, Z.Y., Jia, Y., Hill, D.J., Xu, Y. and Zhang, Y., 2019. Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network. IEEE Transactions on Smart Grid, 10(1), pp.841–851. https://doi.org/10.1109/TSG.2017.2753802.

Lai, G., Chang, W.C., Yang, Y. and Liu, H., 2018. Modeling long- and short-term temporal patterns with deep neural networks. Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pp.95–104. https://doi.org/10.48550/arXiv.1703.07015.

Masood, Z., Gantassi, R., Ardiansyah and Choi, Y., 2022. A Multi-Step Time-Series Clustering-Based Seq2Seq LSTM Learning for a Single Household Electricity Load Forecasting. Energies, 15(7), p.2623. https://doi.org/10.3390/en15072623.

Nti, I.K., Teimeh, M., Nyarko-Boateng, O. and Adekoya, A.F., 2020. Electricity load forecasting: a systematic review. Journal of Electrical Systems and Information Technology, 7, p.13. https://doi.org/10.1186/s43067-020-00021-8.

Oreshkin, B.N., Carpov, D., Chapados, N. and Bengio, Y., 2020. N-BEATS: Neural Basis Expansion Analysis for Interpretable Time Series Forecasting. arXiv preprint arXiv:1905.10437 [cs, stat]. [online] Available at: http://arxiv.org/abs/1905.10437.

Wang, Z., Kong, F., Feng, S., Wang, M., Yang, X., Zhao, H., Wang, D. and Zhang, Y., 2024. Is Mamba effective for time series forecasting? Neurocomputing. [online] Available at: https://doi.org/10.1016/j.neucom.2024.129178.

Qin, Z., Shen, X., Li, D., Sun, W., Birchfield, S., Hartley, R. and Zhong, Y., 2024. Unlocking the Secrets of Linear Complexity Sequence Model from A Unified Perspective. arXiv preprint arXiv:2405.17383. Available at: https://doi.org/10.48550/arXiv.2405.17383.

Appendix 1

Forecast visualisations by model