# Unveiling the Potential of Transformer-Based Models for Efficient Time-Series Energy Forecasting

Imane Moustati * and Noreddine Gherabi

National School of Applied Sciences, Sultan Moulay Slimane University, Khouribga, Morocco
Email: imane.moustati@usms.ac.ma (I.M.); n.gherabi@usms.ma (N.G.)
*Corresponding author

***Abstract*—Accurately forecasting energy consumption is critical in optimizing energy management, reducing costs, and enhancing grid stability. This study uses smart meter data to evaluate the performance of four transformer-based models—Vanilla Transformer, Autoformer, Informer, and SpaceTimeFormer—for energy consumption forecasting. The models are evaluated against statistical benchmarks, with results indicating that Autoformer is the most efficient transformer, achieving the best balance between accuracy and computational complexity, with a Mean Absolute Error (MAE) of 0.540, a Root Mean Square Error (RMSE) of 0.764, a Mean Absolute Percentage Error (MAPE) of 0.091, and an $R^2$ of 0.979. The study focuses on transformer models, establishing their utility for time-series forecasting and identifying Autoformer as the most suitable for this dataset. These findings highlight the transformative potential of advanced architectures for handling complex temporal data and provide a benchmark for future research in energy consumption forecasting.**

***Keywords*—transformers models, Autoformer, time-series forecasting, energy consumption prediction, smart meters**

## I. Introduction

Time series prediction is a vital area of research with broad applications across fields such as physical sciences, environmental studies, finance, healthcare, and energy management [1]. It involves forecasting future variables based on past temporal data and plays a critical role in industries like power generation. The importance of this topic stems from the increasing need for accurate forecasting in various sectors to optimize resource allocation, enhance operational efficiency, and support decision-making processes. In particular, energy consumption forecasting is crucial for balancing energy demand and supply, reducing overproduction, and enhancing grid stability. Accurate short-term predictions enable efficient generator scheduling and long-term strategic planning to minimize costs. However, energy forecasting poses significant challenges due to the irregularity of consumption patterns, seasonal variations, and the presence of uncertainties such as missing data, outliers, and redundancies in the collected data [2]. Traditional forecasting methods struggle to address these complexities, necessitating the development of more advanced predictive models [3]. Recent advancements in technology, such as the widespread deployment of smart meters, have revolutionized energy data collection [4] by providing granular insights into consumption patterns at household levels. This surge in detailed time-series data has paved the way for data-driven solutions to optimize energy usage and predict future demand [5]. Traditionally, deep learning models like Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, Convolutional Neural Networks (CNNs), and their hybrid variants have dominated time series forecasting tasks [6]. However, the sequential nature of RNNs hinders their scalability due to computational inefficiencies, while CNNs require extremely deep architectures to capture long-range dependencies. These limitations highlight the need for more efficient architectures capable of handling long-sequence data effectively. Transformers, initially designed for Natural Language Processing (NLP) tasks, have emerged as a promising alternative for time series forecasting [7]. Unlike RNNs, which process sequences element by element, transformers leverage self-attention mechanisms to process entire sequences simultaneously. This enables transformers to capture long-distance dependencies in time series data more naturally while preserving the positional information of input features through positional encodings [8]. Although originally tailored for NLP, transformers have been adapted for time series applications through various model variations, such as Informer, Autoformer, and SpaceTimeFormer, which aim to address challenges like quadratic memory complexity and scaling for long input sequences [9–11]. These advancements have demonstrated the versatility of transformers in producing competitive results across domains, including time series forecasting [12]. This research paper presents a comprehensive evaluation of four transformer models—Vanilla Transformer, Autoformer, Informer, and SpaceTimeFormer—for

Manuscript received December 26, 2024; revised January 23, 2025; accepted February 11, 2025; published May 9, 2025.

energy consumption forecasting, alongside a benchmark comparison with two RNN-based models to offer additional insights. Using time-series data gathered from smart meters, the study assesses the performance of each model across multiple statistical metrics to determine the most effective approach. This research presents a complete evaluation of transformer models versus proven RNN-based methodologies within the specific context of energy consumption forecasts, and specific research questions leading this work include How do transformer models compare to standard RNN-based models in terms of predicting accuracy for energy consumption; What are the strengths and limits of each model when applied to this dataset. This study offers valuable insights into which architectures are most effective for predicting energy demand by assessing multiple statistical metrics across different models. The findings offer valuable insights into the applicability of transformer models in this domain and will assist smart grid operators, utility firms, and energy providers in selecting optimal forecasting models tailored to their specific operational needs.

The remainder of this paper is organized as follows: Section II provides a review of the existing literature on energy consumption forecasting models. Section III describes the dataset, outlines the preprocessing steps, and explains the methodology, including the models' architecture, configurations and evaluation criteria. Section IV presents the experimental results along with their analysis. Finally, Section V summarizes the key findings and explores potential directions for future research.

## II. Literature Review

Multiple approaches have already been used to predict time-series data in energy consumption applications. Authors in this study offered a system that uses an LSTM-based model to predict energy consumption and influence citizens' consumption behavior [13], and in this research, Lin *et al.* [14] have proposed forecasting energy consumption by combining LSTM and an attention mechanism. They employed the attention method to process data in the training phase and assign weights to data sequences in the input so that the LSTM network could concentrate on the correct sequence segment. Applying this model, they learned efficiently the pattern of electricity changes and improved the prediction's accuracy. In Oliveira's study [15], they presented a reshaped multi-head transformer architecture by focusing on a multi-variable time-series to efficiently predict the buildings' electricity consumption by learning to weigh the attention matrix of features. They offered an interesting performance when comparing the modified vanilla transformer-based model performance with other RNN models. Sparse transformers have also been used in forecasting energy consumption data. Chan and Yeo [7] proposed a sparse transformer-based method to predict electricity load at the household and city levels. Their approach achieved similar accuracy to the state-of-the-art RNN based method while surpassing it in speed by 5 times. In another work, Chan and Yeo [6] also proposed an approach for predicting time series data based on sparse transformers. They offered a model that achieved an accuracy similar to the state of art methods, on the London Smart Meter dataset while outperforming it by 10 times in inference speed. SpaceTimeFormer has also been used to predict house load consumption data in Paris, France [16]. They proposed a self-attention-based model to capture long term dependencies and predict long term sequences of data. Saoud *et al*. [2] offered a hybrid model to predict residential power consumption based on the Stationary Wavelet Transform (SWT), the transformer architecture, and the Time2vec model. They predicted the local features of the energy consumption with SWT and modeled the local trends with the transformer to predict the next wavelet subband. The vanilla transformer has been used in combination with LSTM to accurately predict the time series data of climate change from 2015 to 2018 in Spain [17]. The Autoformer model was employed to offer a robust model that predicts 15 min and 1h load data of the Pecan Street dataset in Texas [18]. And to offer a novel approach to predict energy consumption that can adjust the original sequence according to the actual situation while analyzing the impact of working days and seasonal changes on the electricity demand in Taixing City and Ne South Wales [19]. Li *et al.* [20] proposed a model combining the Autoformer transformer architecture and the Time2Vec model. They tested their model on a monthly total electricity consumption dataset of China from 2009 to 2020. They utilized the Time2vec model to enhance the transformer performance, by effectively embedding the month sequences in the encoder-decoder architecture of the Autoformer. In different research, the Informer model was employed. For example, Xu *et al.* [21] have built an Informer based model to predict power load in Nanchang, and found that it has advantages over cyclic NNs. The multi-step predicting model used seq2seq structure along sparse self-attention mechanism. Li *et al* [22] proposed a hybrid model, that was tested on the BDG2 dataset, where Informer is combined with the Ensemble Empirical Modal Decomposition (EEMD), and the Particle Swarm Optimization (PSO) Algorithm is used to optimize the model's parameters. Time-series data have also been predicted using other techniques. For example, in this study, the author proposed using Stochastic Configuration Networks (SCNs) to predict traffic flow [23], while others offered an adaptive spatio-temporal graph multi-attention network intended for intelligent forecasting of time-series data with intricate spatio-temporal features [24].

## III. Materials and Methods

### A. Dataset and Data Preprocessing

The dataset used in this paper contains smart meter data. The energy use data in KW of around 6,000 homes and companies in Ireland during 2009 and 2010 was gathered once every thirty minutes, each day of the week, by the Irish Smart Metering Project [25]. Hourly compilations of the time series data were made, and any missing or null values were eliminated. After creating input characteristics

at comparable ranges using a min-max normalization into the [0, 1] interval, the K-Means approach is used to cluster data into groups based on typical electrical tendencies. The clustering process plays a crucial role in enhancing model performance by, for example, guiding model selection, reducing noise and improving feature representation. Clustering helps to group similar consumption profiles, which can reduce noise and variability within each cluster. By training models on more homogenous subsets of data, we mitigate the effects of outliers and extreme values that could skew predictions. Each cluster can be treated as a unique subset with its characteristics, enabling models to learn specific patterns that are relevant to that group. Also, the clustering process allows for a more targeted feature representation. Each cluster can be treated as a unique subset with its characteristics, enabling models to learn specific patterns that are relevant to that group. The selected models are then applied to a cluster to forecast energy consumption. The models have some similar parameters (200 epochs, a batch size of 64, a sequence size of 10, and a dropout rate of 0.25). The dataset was divided into 80% training and 20% test samples, with 10,272 observations in the training set and 2,568 in the testing one.

### B. Methodology Workflow

In this section we examine the workflow of the proposed methodology to forecast energy consumption. By implementing four transformer models, we evaluate their performance and highlight the efficient model. Fig. 1 illustrates the process we followed.

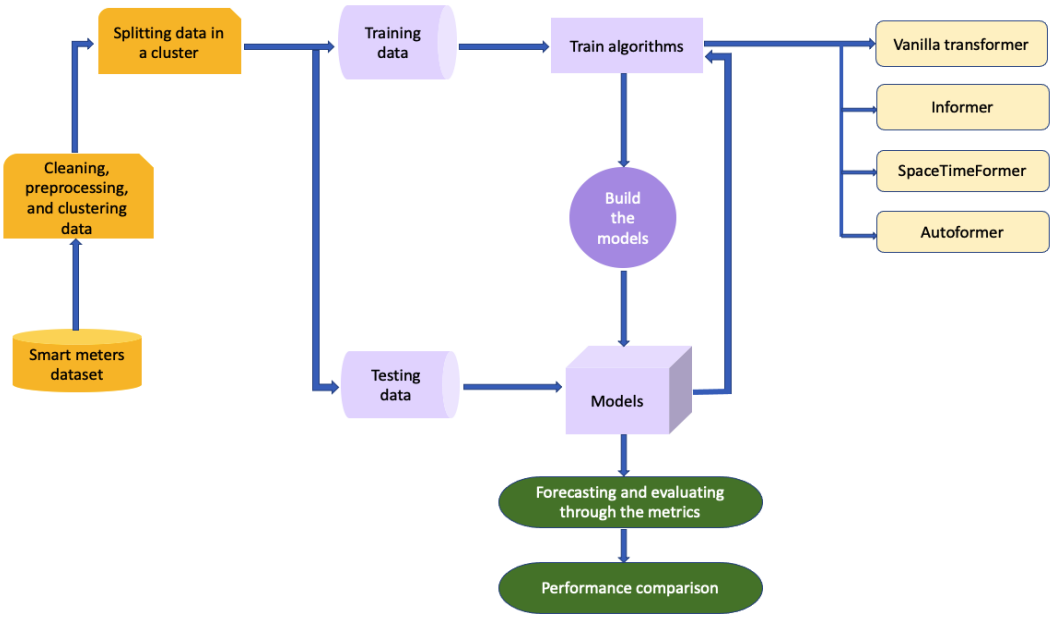



Fig. 1. The methodology workflow for energy consumption forecasting.

### C. The Transformer Models

#### 1) Vanilla transformer

The transformer receives the input sequence for the encoder and decoder layers as well as the decoder layer's output sequence during the training phase. Fig. 2 and 3 depict the structure of a transformer. The input sequence initially passes through a linear layer, with d_model serving as the output size and the number of features serving as the input size. The transformer can process the items of a sequence in parallel, thus increasing the speed of training compared to an LSTM and effectively utilizing the GPU. However, the transformer uses positional encoding to explain the order of each item in the sequence, since data are not sequentially processed. The positional encoding layer generates a matrix with rows of length n and columns of length d_model. The sine and cosine functions are used, respectively, to determine the even and odd values for each item. The dimension index (i) and the current location (k) are represented respectively in Eqs. (1) and (2).

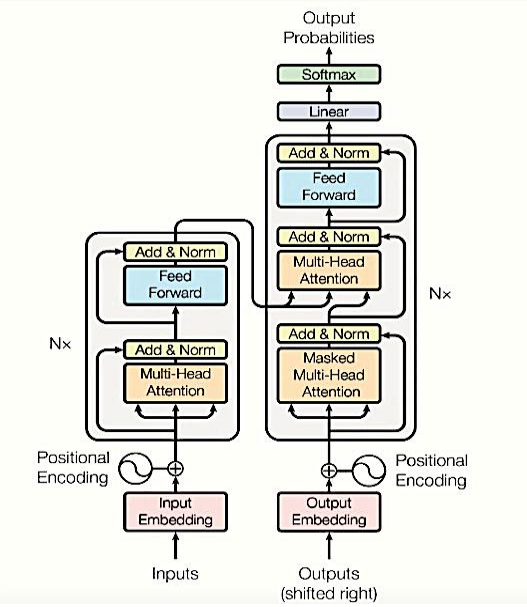



Fig. 2. Structure of a transformer [7].

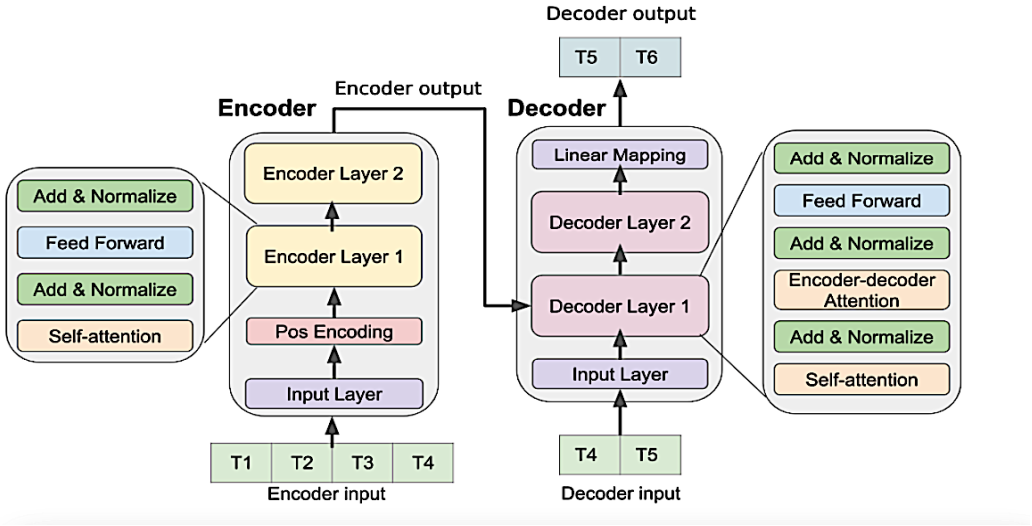



Fig. 3. The structure of a transformer with two encoder-decoder blocks.

$$P(k, 2_i) = \sin\left(\frac{k}{n_i^{2/\,\text{d\_model}}}\right) \quad (1)$$

$$P(k, 2_i + 1) = \cos\left(\frac{k}{n_i^{2/\,\text{d\_model}}}\right) \quad (2)$$

This new matrix is fed to an attention block where it is duplicated three times; each matrix goes to a different linear layer which outputs three similarly structured vectors of parameters: value (*V*), key (*K*) and query (*Q*). If multi-head attention is performed, each vector is split into *p* equal parts, and a dot product between *Qp* and *Kp* is performed, forming QK.

$$\text{Attention scores} = Q \cdot K \quad (3)$$

This newly formed matrix explains the relationship that all the items in the sequence have with each other. The values in the matrix are called the attention scores. This attention score matrix goes through a SoftMax activation function, transforming the attention scores into weights between 0 and 1. These weights are then divided by the square root of d_model and multiplied with *V*.

The higher the weight, the more important the item is for predicting the output sequence.

$$\text{Attention} = \text{SoftMax}\left(\frac{\text{Attention scores}}{\sqrt{\text{d\_model}}}\right)V \quad (4)$$

After multiplying each of the several heads of the QK and V values, they may be concatenated once again to create a single matrix, which will be given to a linear layer after normalization. With the exception of the input sequence, the decoder block's first stages are identical to those of the encoder block. Similar to the encoder part, a linear layer is applied to the input before the positional encoder is performed. The sequence then enters an attention block, during which the output is normalized. Then comes the encoder-decoder attention block. The query, key, and value matrix are expected as input. The decoder block provides the value matrix, while the encoder block provides the query and key matrices. The output of the third attention block is going to be normalized before being passed into a linear layer. And a SoftMax activation function will be employed depending on whether the task is regression or classification. Similar to a typical feed-forward neural network, the output is compared to the correct sequence, and the errors are propagated backward.

*2) Informer*

According to recent research, transformers outperform RNN-type models in terms of expressing long-distance dependencies. Transformers, on the other hand, have three challenges: the quadratic complexity of computation of the self-attention mechanism, high memory utilization, and poor inference in anticipating long-term outcomes. As a result, the developers of Informer [9] improved the transformer model to make it better computationally, memory, and architecturally proficient while retaining stronger predictive capability. As shown in Fig. 4, it is an encoder-decoder structure, with the self-attention distillation mechanism positioned in the encoder layer. Informer produces a sparse attention using the keys and crucial queries. The attentiveness scores show long-tailed distributions between key and critical queries. In reality, the majority of scores are low, with only a handful being high. Then, Informer concentrates on modeling those with significant attention, while the remainder are neglected. This approach generates an organization of sparse attentions, which significantly enhances computing efficiency. Informer additionally brings in self-attention distillation across every two levels of the transformer architecture. A convolution technique is performed to cut the sequence in half, which greatly lowers the training overhead. Informer uses a method at the decoder stage to forecast the results of numerous time steps at once, which can help to lessen the cumulative error issue. Informer developed the notion of sparse bias, using the LogSparse mask approach [26], and dramatically lowered the computing cost of the classic transformer model from O(L2) to O(L log L).

*3) Autoformer*

Autoformer [10] is an improved version of transformer [8] that optimizes it for time series cases, with a focus on managing sophisticated temporal patterns and

overcoming the bottlenecks of compute efficiency and information consumption. Fig. 5 depicts its architecture [10]. Autoformer decomposes trend items into seasonal things and extracts seasonal elements using the moving average approach.

The trend items of every window are produced by computing the mean value of the windows in the initial input time series, and thus having the trend items for the whole series. Also, the seasonal term can be calculated by using the addition model to subtract the trend from the initial input sequence. Using its own internal operators, Autoformer can successfully distinguish a variable's general trend from the projected hidden variables. Its core consists of the series decomposition block module and an improved auto-correlation Process for multi-head attention, which allows it to attain an O(L log L) complexity. It incorporates the decomposition as a built-in block into the deep forecasting model and creates the auto-correlation process to identify period-based connections and aggregate comparable sub-series from underpinning periods.

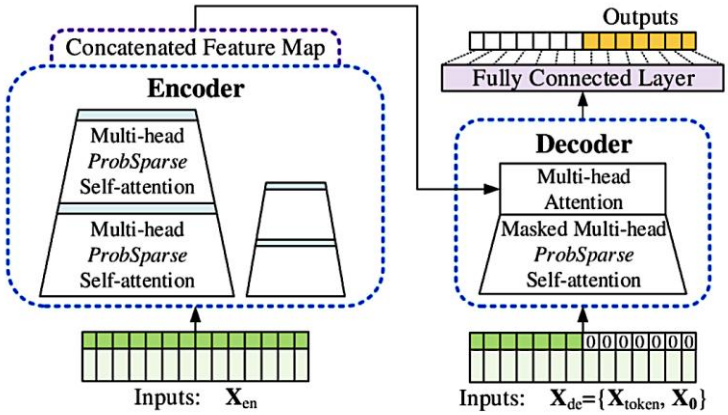



Fig. 4. The Informer model's architecture [8].

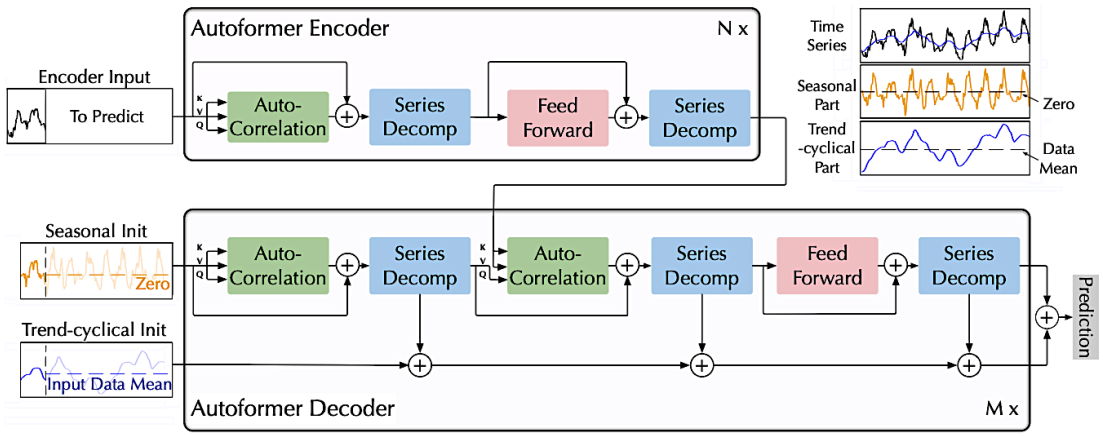



Fig. 5. The Autoformer model's architecture.

*4) SpaceTimeFormer*

SpaceTimeFormer [11] lacks the iterative structure of an RNN, hence it is permutation invariant, and altering the order of the input data has no influence on the end result. The data for the multivariate time series forecasting job is encoded and decoded using a combination of local and global attention. To better discriminate between input from distinct variables in distinct time steps, the model separates the transformer's embedding layer into time embedding and variable embedding. The encoder can choose any segment from the training set as its data. The model uses time2vec to encode information in the time dimension and the anticipated data's relative position, as well as to learn seasonal properties. The relative position information is the location of the current data point over the whole coding sequence, whereas the time information primarily consists of certain calendar characteristics like the year, month, hour or day. For further operations, the output of the time embedding appears in the transformer model's input dimension. The model can discriminate between distinct time steps because of time embedding; however, it still cannot distinguish between different variables. The model must also differentiate between various features within the same sequence as all of the features are in an identical sequence.

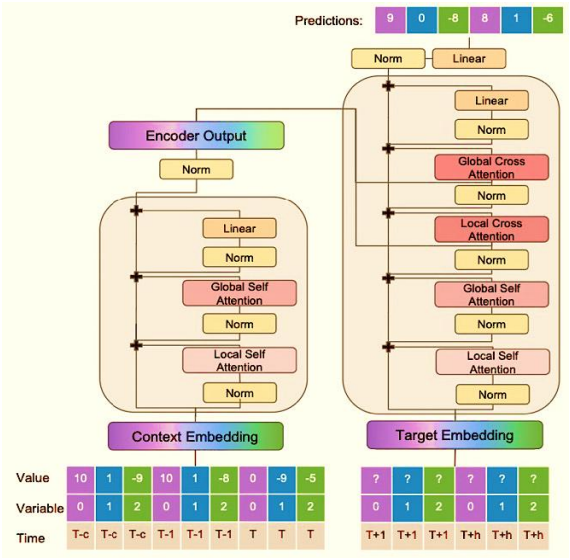



Fig. 6. The SpaceTimeFormer model's architecture.

To identify which time-series data are derived from which initial variables in this situation, an embedding must be included for each variable. In DL, this calculation is processed simply as a standard embedding layer. The SpaceTimeFormer architecture is shown in Fig. 6 [11].

### D. Complexity Comparison

The computational complexity of different models is a critical factor when applying transformer-based architectures to TSF. In particular, the self-attention mechanism in transformers, which allows models to capture long-range dependencies, can be computationally expensive, especially for long sequences.

To address this, recent advancements such as Informer and Autoformer have introduced sparsity and series decomposition mechanisms to improve efficiency. With N as the input sequence length and d as the hidden dimension, Table I compares the algorithmic complexity of several transformer-based models (Vanilla Transformer, Informer, Autoformer, and SpaceTimeFormer) as well as recurrent models (LSTM and RNN), providing a detailed breakdown of their self-attention and overall layer complexities. These complexities highlight the trade-offs between model expressiveness and computational efficiency, which are essential considerations when scaling models to large time-series datasets.

TABLE I. THE COMPLEXITY OF POPULAR MODELS APPLIED TO TSF

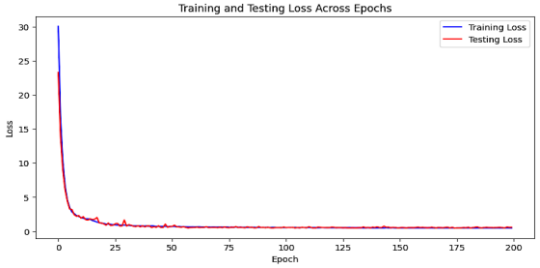
| Model | Self-Attention Complexity | Overall Training Complexity (Per Layer) | | Notes |
|---|---|---|---|---|
| | | Time | Memory | |
| Vanilla Transformer | $O(N^2 \cdot d)$ | $O(N^2 \cdot d)$ | $O(N^2 \cdot d)$ | Has quadratic complexity due to self-attention. |
| Informer | $O(N \cdot \log N \cdot d)$ | $O(N \cdot \log N \cdot d)$ | $O(N \cdot \log N \cdot d)$ | Sparse attention in Informer reduces complexity to logarithmic time, making it efficient for long time-series. |
| Autoformer | $O(N \cdot d)$ | $O(N \cdot d)$ | $O(N \cdot d)$ | Autoformer uses series decomposition and has linear complexity, making it highly efficient for time-series. |
| SpaceTimeFormer | $O(N^2 \cdot d)$ | $O(N^2 \cdot d)$ | $O(N^2 \cdot d)$ | Similar to the Vanilla Transformer but handles spatiotemporal data. Complexity remains quadratic. |
| LSTM | No attention | $O(N \cdot d^2)$ | $O(N \cdot d^2)$ | The LSTM processes sequences step by step. Its complexity depends on the sequence length and hidden units, with quadratic complexity in the hidden dimension. |
| RNN | No attention | $O(N \cdot d^2)$ | $O(N \cdot d^2)$ | Similar to LSTM in complexity but lacks the gating mechanisms, which makes it generally less effective for long sequences. |

## IV. RESULT

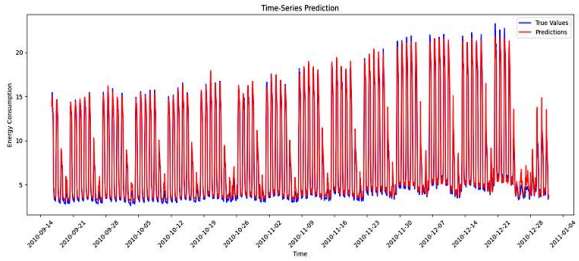
In this section, we predict the cluster's hourly energy consumption. We conducted an evaluation through different lenses to assess the transformer model with the highest performance in forecasting energy consumption. The results are presented in Table II, highlighting the values for the performance metrics discussed earlier ($R^2$, MAPE, RMSE, MAE). We added the performance results of two RNN-based models tested on the same dataset to enhance and enrich the comparison analysis. A vanilla LSTM model performance [5] and a Lag-Augmented LSTM (LA-LSTM) model results [13]. Figs. 7–10 illustrate each model's performance by plotting the training and testing loss across epochs and the time-series graph to visualize the difference between the predicted and the actual values and analyze the differences in performance between the studied transformer models. This combination of metrics is used to emphasize the importance of a multifaceted evaluation approach in assessing the predictive capabilities of the models accurately.

TABLE II. STATISTICAL METRICS' VALUES OF THE COMPARED MODELS

| Model | MAE | RMSE | MAPE | $R^2$ |
|---|---|---|---|---|
| Vanilla Transformer | 0.574 | 0.946 | 0.098 | 0.964 |
| Informer | 0.658 | 1.147 | 0.100 | 0.952 |
| Autoformer | 0.540 | 0.764 | 0.091 | 0.979 |
| SpaceTimeFormer | 1.149 | 1.478 | 9.733 | 0.921 |
| LSTM | 1.599 | 2.106 | 0.195 | 0.841 |
| LA-LSTM | 0.359 | 0.492 | 0.06 | 0.986 |

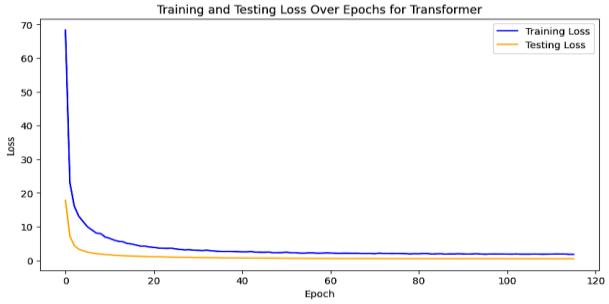



(a)

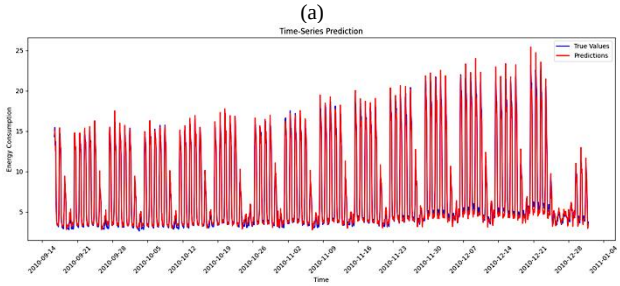



(b)

Fig. 7. Visualizing the Autoformer performance: (a) Train and test loss across epochs; (b) The time-series plot of the predicted and actual data values.

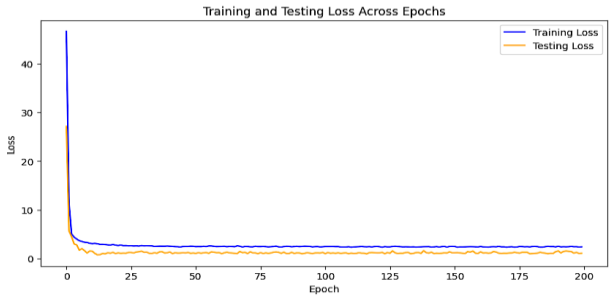



(a)

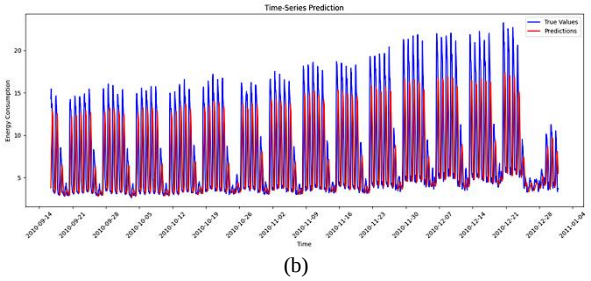



(b)

Fig. 8. Visualizing the Vanilla transformer performance: (a) Train and test loss across epochs; (b) The time-series plot of the predicted and actual data values.

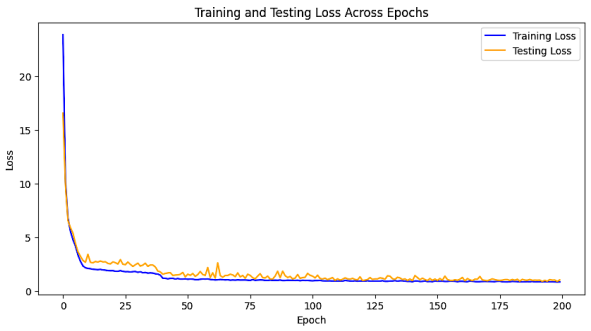



(a)

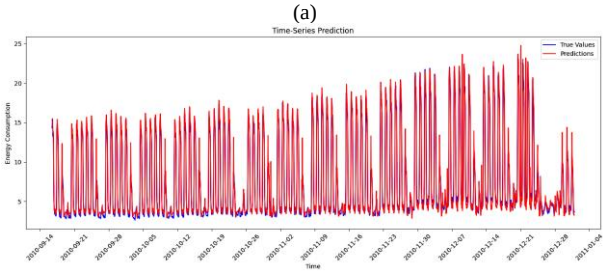



(b)

Fig. 7. Visualizing the SpaceTimeFormer performance: (a) Train and test loss across epochs; (b) The time-series plot of the predicted and actual data values.




(a)




(b)

Fig. 8. Visualizing the Informer performance: (a) Train and test loss across epochs; (b) The time-series plot of the predicted and actual data values.

## V. Discussion

The evaluation metrics in Table II highlight remarkable differences in the performance of the tested models on the energy consumption forecasting task. Among the transformer models trained and tested in this study, the Autoformer demonstrated superior efficiency, achieving the lowest MAE (0.540), RMSE (0.764), and MAPE (0.091) while yielding the highest $R^2$ value (0.979). This suggests that Autoformer is particularly well-suited for capturing both short-term and long-term temporal dependencies in this dataset, outperforming other transformer-based approaches. The Vanilla Transformer, with an $R^2$ of 0.964, also performed well, showcasing its robustness in time-series forecasting. Its metrics (MAE: 0.574, RMSE: 0.946, MAPE: 0.098) indicate a strong balance between accuracy and computational simplicity. However, its performance was slightly behind Autoformer, which is likely due to the latter's specialized mechanisms for multi-scale temporal pattern extraction, and suggests that it slightly struggles with decomposing complex temporal structures in the data. The Informer, while exhibiting decent performance (MAE: 0.658, RMSE: 1.147, MAPE: 0.100, $R^2$: 0.952), lagged behind both the Vanilla Transformer and Autoformer. Its suboptimal performance might stem from challenges in handling this dataset's characteristics, such as seasonality or non-stationarity. SpaceTimeFormer, with the lowest $R^2$ among transformers (0.921), struggled in this context. Its relatively higher MAE (1.149), RMSE (1.478), and MAPE (9.733) suggest it was less effective at capturing the dataset's temporal patterns, potentially due to the added complexity in integrating spatial and temporal dependencies. Comparing the transformer models to the LSTM-based approaches, the LA-LSTM achieved remarkable results (MAE: 0.359, RMSE: 0.570, MAPE:

0.045, $R^2$: 0.986), outperforming all other models, including the Autoformer. These results emphasize the impact of augmenting the LSTM architecture with lagged windows of past observations, allowing it to excel in identifying temporal dependencies specific to this dataset, and revealed that while transformer-based models generally outperform LSTM in direct implementations, augmenting LSTM with lagged inputs can result in substantial improvements, outperforming even the best transformer models on this dataset. The LA-LSTM's enhancement with lagged inputs further amplifies LSTM networks' capability, allowing it to maintain relevant information across longer sequences than traditional LSTMs. While transformer architectures like Autoformer are designed for broader applications and can handle various types of data effectively, the specific enhancements made in LA-LSTM for time series data allow it to outperform these models in scenarios where historical context plays a crucial role. If the dataset exhibits strong autocorrelation or seasonal patterns, LSTM enhancements can provide superior predictive power. Therefore, for datasets where historical values significantly influence future outcomes, such as financial or environmental data, LA-LSTM is likely a better choice, and for tasks that benefit from parallel processing and extensive contextual understanding across longer sequences (e.g., multi-step forecasts), transformers may be more advantageous. Thus, LA-LSTM can outperform transformer models under specific conditions related to dataset characteristics and forecasting requirements. The Vanilla LSTM, however, was significantly outperformed by all transformer models and LA-LSTM, suggesting that while LSTMs are effective for many sequence-to-sequence tasks, their reliance on gated recurrent units and sequential processing makes them less efficient in capturing the global temporal dependencies that transformer-based models handle more effectively. This reinforces the importance of either advanced architectural enhancements or leveraging the latest transformer-based innovations for handling complex temporal data. This study underscores the efficacy of transformer models, particularly the Autoformer, for time-series forecasting in energy consumption datasets, and the results emphasize the critical role of architectural innovation in improving time-series forecasting accuracy. While LA-LSTM offered the best overall performance and demonstrated the value of combining recurrent architectures with lagged features to enhance forecasting accuracy, its results serve as a benchmark from existing literature rather than a primary focus of this research. The findings establish transformer models as state-of-the-art solutions for time-series forecasting, with Autoformer emerging as the most efficient transformer for this dataset, due to its decomposition-based design, which aligns well with the periodic and trend-based structure of energy consumption data.

## VI. Conclusion

This study presents a complete evaluation of transformer models versus proven RNN-based methodologies within the specific context of energy consumption forecasts. It conducted a comprehensive evaluation of transformer-based models, focusing on their ability to handle complex temporal dependencies in time-series data. The outcomes of this research are particularly significant for smart grid operators, utility firms, and energy providers who aim to improve the precision of their energy demand predictions. Among the transformer models tested, Autoformer emerged as the most effective, achieving the best performance across all metrics (an MAE of 0.540, an RMSE of 0.764, a MAPE of 0.091, and an $R^2$ equal to 0.979). Its decomposition-based design proved adept at capturing both short-term fluctuations and long-term trends, making it particularly suitable for the energy consumption dataset used. The Vanilla Transformer also demonstrated strong performance, reflecting its versatility and computational efficiency, though it fell slightly behind Autoformer in handling intricate temporal patterns. The Informer and SpaceTimeFormer models showed varying levels of success, with SpaceTimeFormer struggling the most, possibly due to the additional complexity of integrating spatial and temporal dependencies, which may not have been as critical in this dataset. These findings underscore the transformative potential of transformer models for time-series forecasting in energy-related applications, and establish transformer models, especially Autoformer, as state-of-the-art solutions for time-series forecasting. The results highlight the critical role of architectural advancements, such as decomposition and attention mechanisms, in enhancing forecasting performance for complex datasets like energy consumption. This work paves the way for further exploration into transformer-based architectures, encouraging their adoption in real-world forecasting scenarios. The contributions of this study can further be enhanced in future research. Future work could focus on strengthening predictive performance by developing advanced hybrid models, such as integrating LA-LSTM with Autoformer to boost accuracy and robustness. Also, we can explore methods to enhance the model's interpretability by applying Explainable AI (XAI) techniques and addressing the models' generalizability across different datasets.

## Conflict of Interest

The authors declare no conflict of interest.

## Author Contributions

Conceptualization, methodology, software, formal analysis, data curation and writing—original draft preparation, Imane Moustati; writing—review and editing, validation, supervision and project administration, Noreddine Gherabi. All authors had approved the final version.

## References

[1] I. Moustati, N. Gherabi, and M. Saadi, "Leveraging the internet of behaviours and digital nudges for enhancing customers financial decision-making," *International Journal of Computer Applications*

*in Technology*, vol. 1, no. 1, 2024. doi: 10.1504/IJCAT.2024.10065772
[2] L. S. Saoud, H. Al-Marzouqi, and R. Hussein, "Household energy consumption prediction using the stationary wavelet transform and transformers," *IEEE Access*, vol. 10, pp. 5171–5183, 2022. doi: 10.1109/ACCESS.2022.3140818
[3] J. Zhang, H. Zhang, S. Ding, and X. Zhang, "Power consumption predicting and anomaly detection based on transformer and k-means," *Front Energy Res.*, vol. 9, Oct. 2021. doi: 10.3389/fenrg.2021.779587
[4] I. Moustati, N. Gherabi, H. El-Massari, and M. Saadi, "From the Internet of Things (IoT) to The Internet of Behaviors (IoB) for data analysis," in *Proc. 2023 7th IEEE Congress on Information Science and Technology (CiSt)*, IEEE, Dec. 2023, pp. 634–639. doi: 10.1109/CiSt56084.2023.10409989
[5] I. Moustati, N. Gherabi, and M. Saadi, "Time-series forecasting models for smart meters data: An empirical comparison and analysis," *Journal Européen des Systèmes Automatisés*, vol. 57, no. 05, pp. 1419–1427, Oct. 2024. doi: 10.18280/jesa.570517
[6] J. W. Chan and C. K. Yeo, "Electrical power consumption forecasting with transformers," in *Proc. 2022 IEEE Electrical Power and Energy Conference (EPEC)*, IEEE, Dec. 2022, pp. 255–260. doi: 10.1109/EPEC56903.2022.10000228
[7] J. W. Chan and C. K. Yeo, "A transformer based approach to electricity load forecasting," *The Electricity Journal*, vol. 37, no. 2, 107370, Mar. 2024. doi: 10.1016/j.tej.2024.107370
[8] A. Vaswani *et al.*, "Attention is all you need," arXiv preprint, arXiv:1706.03762, Jun. 2017.
[9] H. Zhou *et al.*, "Informer: Beyond efficient transformer for long sequence time-series forecasting," arXiv preprint, arXiv:2012.07436, Dec. 2020.
[10] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: decomposition transformers with auto-correlation for long-term series forecasting," arXiv preprint, arXiv:2106.13008, Jun. 2021.
[11] J. Grigsby, Z. Wang, N. Nguyen, and Y. Qi, "Long-range transformers for dynamic spatiotemporal forecasting," arXiv preprint, arXiv:2109.12218, Sep. 2021.
[12] Q. Wen *et al.*, "Transformers in time series: A survey," arXiv preprint, arXiv:2202.07125, Feb. 2022.
[13] I. Moustati, N. Gherabi, and M. Saadi, "Building an IoB ecosystem for influencing energy consumption in smart cities," *Data and Metadata*, vol. 3, Oct. 2024. doi: 10.56294/dm2024.441
[14] Z. Lin, L. Cheng, and G. Huang, "Electricity consumption prediction based on LSTM with attention mechanism," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 15, no. 4, pp. 556–562, Apr. 2020. doi: 10.1002/tee.23088
[15] H. S. Oliveira and H. P. Oliveira, "Transformers for energy forecast," *Sensors*, vol. 23, no. 15, 6840, Aug. 2023. doi: 10.3390/s23156840
[16] G. Sreekumar, J. P. Martin, S. Raghavan, C. T. Joseph, and S. P. Raja, "Transformer-based forecasting for sustainable energy consumption toward improving socioeconomic living: AI-enabled energy consumption forecasting," *IEEE Syst Man Cybern Mag*, vol. 10, no. 2, pp. 52–60, Apr. 2024. doi: 10.1109/MSMC.2023.3334483
[17] H. Wang, J. Li, and L. Chang, "Predicting time series energy consumption based on transformer and LSTM," in *Proc. International Conference on 6GN for Future Wireless Networks*, 2024, pp. 299–314. doi: 10.1007/978-3-031-53401-0_27
[18] Y. Jiang *et al.*, "Very short-term residential load forecasting based on deep-autoformer," *Appl. Energy*, vol. 328, 120120, Dec. 2022. doi: 10.1016/j.apenergy.2022.120120
[19] Z. Wang, Z. Chen, Y. Yang, C. Liu, X. Li, and J. Wu, "A hybrid Autoformer framework for electricity demand forecasting," *Energy Reports*, vol. 9, pp. 3800–3812, Dec. 2023. doi: 10.1016/j.egyr.2023.02.083
[20] X. Li, Y. Zhong, W. Shang, X. Zhang, B. Shan, and X. Wang, "Total electricity consumption forecasting based on Transformer time series models," *Procedia Comput Sci*, vol. 214, pp. 312–320, 2022. doi: 10.1016/j.procs.2022.11.180
[21] H. Xu, Q. Peng, Y. Wang, and Z. Zhan, "Power-load forecasting model based on informer and its application," *Energies (Basel)*, vol. 16, no. 7, 3086, Mar. 2023. doi: 10.3390/en16073086
[22] F. Li *et al.*, "Improving the accuracy of multi-step prediction of building energy consumption based on EEMD-PSO-Informer and long-time series," *Computers and Electrical Engineering*, vol. 110, 108845, Sep. 2023. doi: 10.1016/j.compeleceng.2023.108845
[23] Y. Lin, "Long-term Traffic flow prediction using stochastic configuration networks for smart cities," *IECE Transactions on Intelligent Systematics*, vol. 1, no. 2, pp. 79–90, Sep. 2024. doi: 10.62762/TIS.2024.952592
[24] X.-B. Jin, H. Ma, J.-Y. Xie, J. Kong, M. Deveci, and S. Kadry, "Ada-STGMAT: An adaptive spatio-temporal graph multi-attention network for intelligent time series forecasting in smart cities," *Expert Syst Appl*, vol. 269, 126428, Apr. 2025. doi: 10.1016/j.eswa.2025.126428
[25] Commission for Energy Regulation (CER). (2012). CER Smart Metering Project—Electricity Customer Behaviour Trial, 2009–2010 [dataset]. 1st Ed. Irish Social Science Data Archive. SN: 0012-00. [Online]. Available: https://www.ucd.ie/issda/data/commissionforenergyregulationcer/
[26] W. Li and K. L. E. Law, "Deep learning models for time series forecasting: A review," *IEEE Access*, vol. 12, pp. 92306–92327, 2024. doi: 10.1109/ACCESS.2024.3422528