2024 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT) | 979-8-3315-3313-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/3ICT64318.2024.10824647

ACyLeR: An enhanced iTransformer for Long-Term Time-Series Forecasting Using Adaptive Cycling Learning Rate

Mustafa Kamal
Department of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
7025231001@student.its.ac.id
Department of Information Technology
Telkom University, Surabaya Campus
Surabaya, Indonesia
mustafakamal@telkomuniversity.ac.id

Ary Mazharuddin Shiddiqi*
Department of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
ary.shiddiqi@its.ac.id

Ervin Nurhayati
Department of Environmental Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
ervin@its.ac.id

Andika Laksana Putra
Department of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
5025211001@student.its.ac.id

Farrela Ranku Mahhisa
Department of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
5025211129@student.its.ac.id

***Abstract*—Long-term time-series forecasting is critical in numerous domains, including economics, climate modeling, and energy management. Traditional deep learning models often struggle with optimizing hyperparameters, which can lead to suboptimal performance and increased sensitivity to initial conditions. This research addresses the problem by proposing an enhanced iTransformer model that integrates an Adaptive Cycling Learning Rate (ACLR) mechanism, named ACyLeR. The ACLR algorithm dynamically adjusts the learning rate during the training phase for better convergence and generalization while minimizing the risk of overfitting. The experiments were written in Python and tested using univariate Water Supply in Melbourne (WSM) and multivariate exchange rate (ER) datasets with 70% training, 10% validation, and 20% testing data grouping. Experimental results demonstrate that the ACyLeR with ACLR outperforms existing baseline models by achieving lower loss values and higher accuracy. The results significantly advance time-series forecasting using iTransformer.**

Index Terms— iTransformer, Adaptive Cycling Learning Rate (ACLR), Water Demand Forecasting

I. Introduction

Time-series forecasting is critical in numerous fields, including economics, healthcare, and environmental science. Predicting future values based on historical data might inform important decision-making processes [1]–[3]. Accurate forecasting models offer insights for proactive interventions and optimal resource allocation. Traditional methods like ARIMA and exponential smoothing have established the foundation for time-series analysis. However, these approaches have several disadvantages. They often struggle with non-linear patterns and long-term dependencies.

The advent of deep learning has revolutionized this domain to improve the accuracy and the ability to capture complex patterns in data [4]–[6]. Recently developed iTransformer [7] addresses some of these issues by effectively leveraging attention mechanisms to model long-term dependencies. However, deep learning-based models can be computationally expensive and require substantial data. Moreover, these models can be sensitive to hyperparameters. Therefore, finding optimized models' hyperparameters can be challenging [8], [9].

This study aims to introduce an enhanced version of the iTransformer by integrating an adaptive cycling learning rate mechanism to improve its convergence and efficiency with minimal manual tuning [7]. Simultaneously, the robustness and accuracy of the model must also be maintained to adapt to various time-series datasets. We evaluate its performance against existing baseline models across various experimental setups by observing the sensitivity of the ACyLeR to hyperparameters.

This paper is organized as follows: Section II discusses previous research on this topic and the gaps filled by this study. Section III describes the problem definition, architecture, and mechanism of the proposed method used in this study. Section IV presents the experiment analysis and results related to the proposed method. Finally, Section V presents the conclusions of this study.

II. Literature Review

A. Existing Time Series Forecasting Methods

Time series forecasting has a deep history, with traditional methods like ARIMA and exponential smoothing acting as fundamental methodologies [2]. These approaches work well for linear patterns and short-term forecasting, but they frequently fall short when dealing with nonlinear trends and long-term relationships. Machine learning and deep learning have generated models such as Long Short-Term Memory (LSTM) and gated recurrent units (GRU), which can capture complex patterns and dependencies more efficiently [10]–[12].

The Transformer algorithm proposed by Vaswani in 2017 is widely used for time series forecasting [13]. Since then, many research has been conducted to improve the Transformer model. and its many modifications, such as Autoformer [14] that improve forecasting accuracy by utilizing autocorrelation to capture seasonal and trend components, FEDformer [15] reduces computational complexity through frequency-enhanced decomposed attention, Robformer [16] enhances robustness against noise and missing values, and PatchTST [17] The introduction of patch-based processing effectively handles long-term dependencies. Transformer-based model developments improve time-series forecasting accuracy and efficiency.

B. Time Series Forecasting using iTransformer

iTransformer has gained attention for its ability to handle time series data by leveraging attention mechanisms [7]. This model excels in capturing long-term dependencies and modeling complex patterns, making it a powerful tool for time-series forecasting. The fundamental Transformer architecture incorporates a temporal token with a multivariate representation for every time step. iTransformer individually incorporates each series into a distinct variate token. The attention module exhibited multivariate correlations, while the feedforward network encoded the sequential representations [7].

iTransformer is frequently utilized to predict applications across multiple fields. Short-term forecast of photovoltaic power based on weather parameters in Guizhou mountainous areas [18], estimation of the accurate remaining usable life for lithium-ion batteries [19], solar irradiance prediction with multi-meteorological variables [20], an innovative method for predicting the course of vessels, taking into account the dynamic and context-specific factors [21], heart rate prediction [22] and mobile perception dual-level base station sleep control for cellular traffic forecasting [23]. However, one of the issues with the iTransformer, and deep learning models in general, is their sensitivity to hyperparameters, notably the learning rate, which can substantially impact the model's performance and convergence pace.

C. The role of learning rates in model training

Learning rates are important for training neural networks because they influence how fast and effectively a model converges to an ideal solution. Fixed learning rates might result in suboptimal performance since they may be too high or too low at various training phases [24]. Adaptive learning rates, such as those employed in Adam or RMSprop, alter the learning rate during training, enabling gains over fixed learning rates but still encountering constraints when dealing with complicated, i.e. variable datasets [25]. Current research has explored various adaptive learning rate mechanisms, but there remains a gap in integrating these mechanisms effectively with models like the iTransformer.

D. State of the art

The current state of the art in time-series forecasting is distinguished by advanced Transformer based models (Table I). These include Autoformer's use of autocorrelation for seasonal patterns, FEDformer's reduction of computational complexity, Robformer's robustness against noise and missing data, and PatchTST's ability to manage long-term dependencies. Our proposed model, iTransformer with Adaptive Cyclical Learning Rate (ACLR), enhances learning processes through an adaptive cycle learning rate.

TABLE I
Comparison of various Transformer models and their improvements

Model	Authors	Improvement
Transformer	Vaswany et al, 2017 [13]	Widely used for time-series forecasting with multi-head attention and positional embedding
Autoformer	Wu et al, 2021 [14])	Utilizes autocorrelation to capture seasonal and trend components
FEDformer	Zhou et al, 2022 [15]	Reduces computational complexity through frequency-enhanced decomposed attention
Robformer	Yu et al, 2024 [16]	Enhances robustness against noise and missing values
PatchTST	Nie et al, 2022 [17]	Introduces patch-based processing to handle long-term dependencies effectively
iTransformer	Lie et al, 2023 [7]	Inverts embedded tokens and global series aggregation and is better leveraged
iTransformer with ACLR	*Our proposed model*	*Enhances iTransformer with adaptive cycle learning rate to improve model learning processes*

Table I presents several transformer models. Despite the publication of models like Robformer after iTransformer, the latter demonstrates superior performance evaluation across many datasets, including electricity, exchange, and traffic. We choose iTransformer for modification with ACyLeR.

III. Proposed Method

A. Problem Definition

Consider a time-series $X = [x_1, x_2, \ldots, x_T]$ where x_t denotes the time-series component at day t. Our objective is to forecast the value of Y_t, where $t \geq T$, by application of a model f that transforms the input space $X(x_t \in R^d)$ into Y. The number of features encoding an observation x_t at time t is denoted as d, and $f_\vartheta(\cdot)$ is defined as a function of the ACyLer. A variable $y_t = x_{t+\tau}$ represents the future values of the input x_t, with τ denoting the time steps forward. The challenge is to effectively handle the dynamic data distributions, characterized by $P(X_t, Y_t) \neq P(X, Y)_{t+1}$, and these changes (concept drift) are unknown to the model $f_\vartheta(\cdot)$.

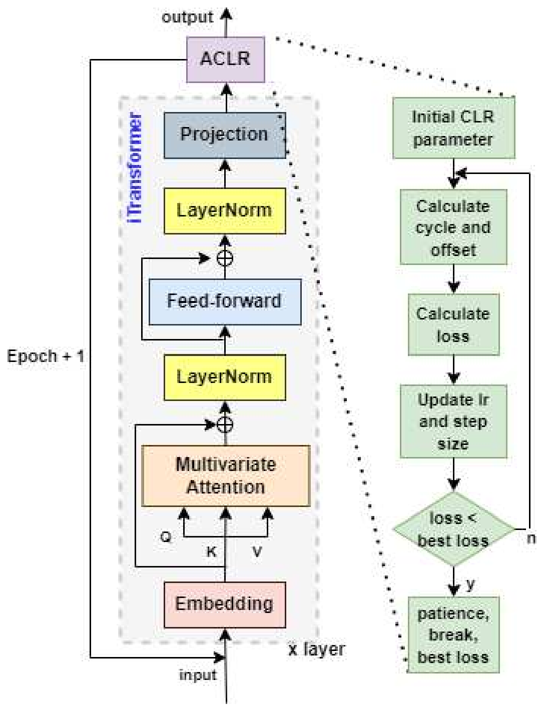

Fig. 1. The architecture of ACyLeR, an enhanced iTransformer with adaptive cycling learning rate mechanism

B. The architecture of the ACyLeR

The ACyLeR architecture builds upon the original Transformer model, incorporating modifications to optimize it for time-series forecasting (Figure 1). The core of the model remains the self-attention mechanism, which allows the model to weigh the importance of different time steps dynamically. However, we introduce several enhancements to improve performance and adaptability.

Our proposed AcyLeR dynamically adjusts the learning rate during training (Figure 1). The algorithm integrates an early stopping mechanism to prevent overfitting while improving model convergence and generalization. The ACLR algorithm adjusts the learning rate based on the model's performance, unlike fixed learning rates that are difficult to optimize and classic cyclical learning rates that aid in avoiding local minima. This adaptive method enhances the learning process by enabling the model to adjust dynamically and achieve superior overall performance.

C. Adaptive cycling learning rate mechanism

Unlike traditional fixed or adaptive learning rates, the cycling learning rate varies periodically within a predefined range (Algorithm 1). This approach helps escape local minima and converge faster to an optimal solution. The learning rate cycle is designed based on the idea that different phases of training benefit from various learning rates. In figure 1, during the initial stages, a higher learning rate facilitates rapid learning, while in later stages, a lower learning rate fine-tunes the model.

We altered the training loop to dynamically update the learning rate based on the epoch and training progress. The lowest and maximum learning rates and the step size determine how frequently the learning rate changes. This integration requires careful tuning to ensure that the learning rate variations align with the model's learning needs.

Mathematically, a sinusoidal function or a triangular waveform can describe the adaptive cycling learning rate, providing smooth transitions between the minimum and maximum learning rates. The learning rate at each epoch can be computed as follows:

$$\mathrm{lr}(t) = \mathrm{lr}_{\mathrm{min}} + 0.5 \times (\mathrm{lr}_{\mathrm{max}} - \mathrm{lr}_{\mathrm{min}}) \times (1 + \cos(\frac{2\pi t}{T})) \quad (1)$$

where $\mathrm{lr}(t)$ is the learning rate at epoch t, $\mathrm{lr}_{\mathrm{min}}$ and $\mathrm{lr}_{\mathrm{max}}$ are the minimum and maximum learning rates, respectively, and T is the total number of epochs in one cycle.

The method initializes the learning rate and parameters and then iterate over each epoch from 1 to *num_epochs*. During each epoch, the algorithm cycled for each batch of the *data* dataset. The algorithm computes the cycle and offsets for the current iteration inside the epoch. Based on the *cycle* and offset, the algorithm calculates an adjusted learning rate *lr* using the *min_lr*, *max_lr*, and *step_size*. Throughout training, the learning rate is cyclically adjusted to study various learning rates.

The program calculated the value of *lr* within a specified range and then compared the current loss with the previous loss. When the loss surpasses a certain threshold, the algorithm reduces the value of the *step_size*. This technique decelerates the rate of learning when the loss experiences a rapid increase. Conversely, if the loss is insignificant, the value of the *step_size* is augmented to expedite the adjustment of the learning rate. The model's parameters are modified according to the determined *lr*. In this stage, the model's weights are modified by taking into account the gradient loss and considering the parameters. The variable *prev_loss* is updated with the current value of the loss and is subsequently utilized in the subsequent iteration.

The program assesses whether the current loss is superior to the *best_loss* achieved. If the loss has been reduced, modify the *best_loss* and set the *best_epoch* to the current epoch. When the loss improves, the counter for *patience* is reset to its highest value. The counter for *patience* will fall as the loss continues. Next, the algorithm verifies whether the value of *patience* has reached zero. In the event of this happening, the training will be halted and the loop will be exited. Once all epochs have been completed or early halting is triggered, the process provides the *model* and the *best_epoch* that corresponds to the lowest loss. The optimal model yields the minimum loss value.

D. Evaluation metrics

The accuracy and stability of the model forecasts were evaluated using the Mean Absolute Error (MAE). see Equation 2, and Mean Squared Error (MSE), see Equation 3, of the evaluation metrics when several forecasting models were employed for the quantitative analysis and evaluation.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{2}$$

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{3}$$

where n is the total number of observations or data points, y_i is the actual value of the i-th observation, and $\hat{y}_i$ is the predicted value of the i-th observation.

IV. EXPERIMENTS

A. Datasets

To evaluate the performance of the ACyLeR, we conducted experiments using two real-world datasets:

- The dataset on exchange rates (ER) was obtained using a panel data consisting of daily exchange rates from 8 nations over the period from 1990 to 2016 [14]. This dataset represents multivariate short-time series data with 7.588 rows.
- Water Supply in Melbourne (WSM), an open public dataset of daily accumulated water usage volumes supplied by the Melbourne water storage systems from ArcGIS. Original WMS data from the open data ArcGIS provide 47180 rows of daily supply data. Data was collected from December 1890 to December 2020. Nevertheless, a computerized dating system was unable to identify water consumption data from the years 1890 to 1899. Hence, we eliminate the 2792 rows of data from the 1800s in the dataset. Consequently, we possess a total of 44388 data entries spanning from January 1, 1900, to December 31, 2020. The information provided represents a collection of long-term time series data for the water supply in Melbourne.

B. Experimental setup

Each dataset was divided into training, validation, and testing sets. The training set trained the models, the validation set tuned the hyperparameters, and the test set evaluated the final performance. Each dataset was divided into three groups: 70% for training, 10% for validation, and 20% for testing. Then, we observe the performance of the proposed ACyLeR using the MAE and MSE. To provide a comprehensive evaluation, we also compare the performance of the ACyLeR with baseline iTransformer models. Each model is trained and evaluated using the same experimental setup to ensure a fair comparison. We also tested the models with three prediction windows, i.e., 96, 192, and 336. This scenario aims to see the effect of the magnitude of improvement on different window conditions.

C. Analysis and results

We evaluate each model's accuracy, convergence speed, and robustness to different datasets. We focus on the impact of the ACLR on the iTransformer's performance compared to the non-adaptive learning rate setup:

Algorithm 1 Training model of the proposed ACyLeR with early stopping

Require: model: The model to train.
Require: data: The dataset to train the model on.
Require: initial_lr: The initial learning rate.
Require: num_epochs: The number of epochs to train for.
Require: patience: The patience for early stopping.

```
min_lr ← initial_lr/10
max_lr ← initial_lr × 10
num_iterations ← len(data)/batch_size {Calculate the total number of iterations per epoch}
step_size ← (max_lr − min_lr)/num_iterations
adaptive_threshold ← 0.95
prev_loss ← ∞ {Initialize with a large value}
best_loss ← ∞
best_epoch ← 0
for epoch ← 1 to num_epochs do
    for iteration ← 1 to num_iterations do
        loss ← calculate_loss(model, batch) {Perform Adaptive Cyclical Learning Rate adjustment}, cycle ← iteration mod 2
        lr ← min_lr + cycle × step_size
        if lr > max_lr then
            lr ← max_lr
        else if lr < min_lr then
            lr ← min_lr
        end if
        if loss > prev_loss × adaptive_threshold then
            step_size ← step_size × 0.5
        else
            step_size ← step_size × 2
        end if
        model.param_update(lr) {Update the model parameters with the adjusted learning rate.}
        prev_loss ← loss {prev_loss adjustment with the last loss for the next looping and early stopping}
        if current loss < best_loss then
            best_loss ← loss, best_epoch ← epoch
            patience ← max(patience, 5) {Reset patience}
        else
            patience ← patience − 1
            if patience == 0 then
                stop
            end if
        end if
    end for
    if patience == 0 then
        stop
    end if
end for
return model, best_epoch =0
```

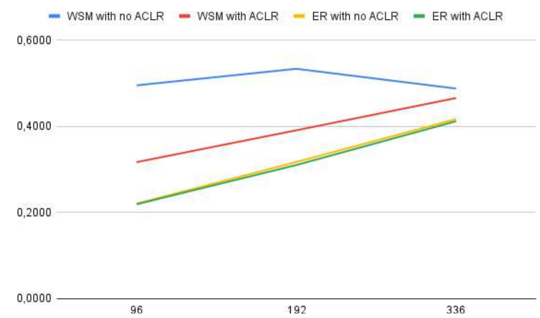

Fig. 2. Plot visualization of MAE evaluation from each model.

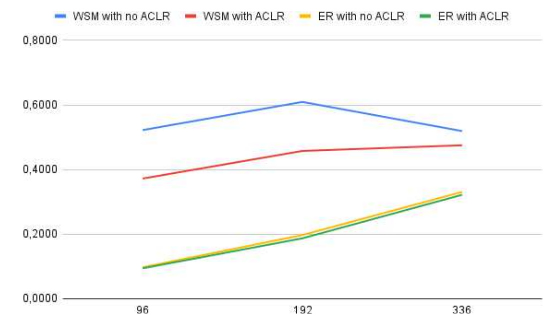

Fig. 3. Plot visualization of MSE evaluation from each model.

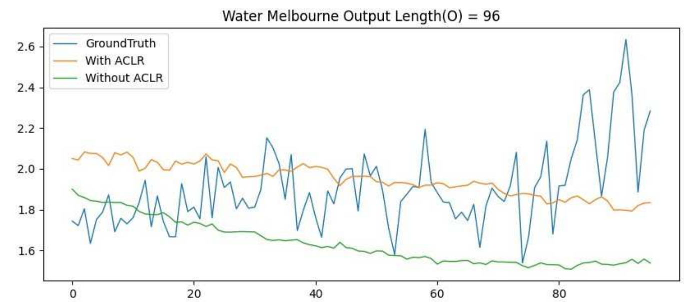

Fig. 4. WSM dataset test results in 96 window length with and without ACyLeR

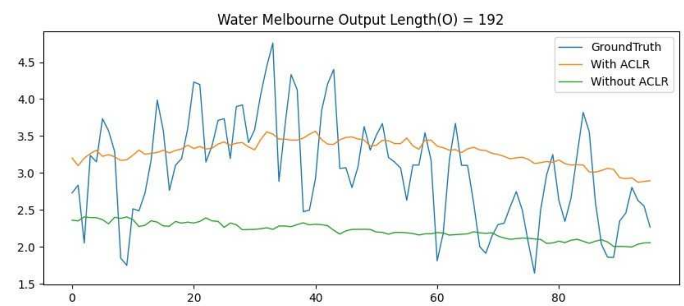

Fig. 5. WSM dataset test results in 192 window length with and without ACyLeR

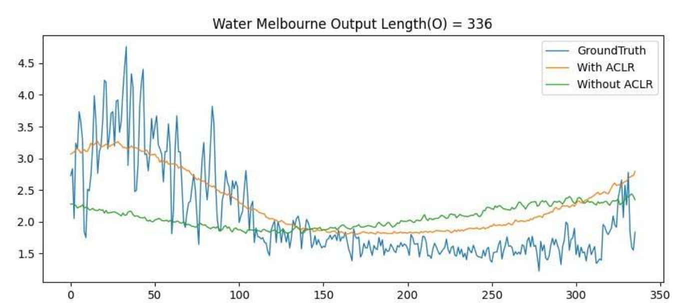

Fig. 6. WSM dataset test results in 336 window length with and without ACyLeR

- Experiment results (figure 2 and figure 3) indicate that the performance peaks at a prediction window of 192 (figure 5), after which it slightly decreases. In contrast, the ACyLeR demonstrates a consistent upward trend as the prediction window length increases on the WSM dataset. Table II shows that the ACyLeR on the WSM dataset produced a lower MAE by more than 0.1 points compared to the iTransformer model. While on the ER dataset, the reduction in the MAE evaluation metric is insignificant at only 0.004. This trend indicates that the ACLR helps maintain or improve model performance over longer prediction windows. While on the ER dataset, both models show a nearly linear increase in performance as the prediction window lengthens, with the ACyLeR providing only a marginal improvement. This suggests that while ACLR has a notable impact on iTransformer performance on the WSM dataset, its effect on the ER dataset is less pronounced.
- The MSE of iTransformer on the WSM dataset initially increases, peaking at a window length of 192 (figure 5) before slightly decreasing at 336 (figure 6). This suggests that while the model's performance worsens as the window length increases to 192, it begins to recover slightly for longer windows. On the other hand, the ACyLeR model shows a steady increase in MSE on the same dataset, indicating a more consistent but slightly less effective performance than the iTransformer. The MSE also increased for both models, as seen in Table II. The ACyLeR produced better MSE on the WSM dataset by more than 0.12 points compared to the iTransformer model. While on the ER dataset, the reduction in the MSE evaluation metric is insignificant at only 0.007. Both models exhibit a nearly identical pattern on the ER dataset. The MSE increases as the prediction window lengthens, showing a significant jump from 96 to 192, after which the increase becomes more gradual on extended window sizes. The slight improvement produced by ACyLeR suggests a marginal benefit, but the difference is not substantial.

The results of this study (figure 4, 5, 6)indicate that the use of ACyLeR results in a more significant improvement in the WSM dataset with long-time series univariate data compared to ER with short-time series with multivariate data.

TABLE II
MODEL RESULTS USING MAE AND MSE METRIC EVALUATION

Dataset	Prediction Window	MAE		MSE	
		iTransformer	ACyLeR	iTransformer	ACyLeR
Water Supply in Melbourne (WSM) (Univariate)	96	0.4956	**0.3172**	0.5222	**0.3721**
	192	0.5340	**0.3912**	0.6094	**0.4577**
	336	0.4881	**0.4662**	0.5192	**0.4752**
	Average	0.5059	**0.3915**	0.5503	**0.4350**
Exchange Rate (ER) (Multivariate)	96	0,2210	**0.2195**	0.0974	**0.0943**
	192	0.3176	**0.3106**	0.1973	**0.1871**
	336	0.4172	**0.4123**	0.3308	**0.3219**
	Average	0.3186	**0.3141**	0.2085	**0.2011**

V. CONCLUSION

This research presents an enhanced version of the iTransformer for time-series forecasting, incorporating an adaptive cycling learning rate mechanism. We aim to address the limitations of existing models, particularly the sensitivity to hyperparameters and the challenges in optimizing learning rates for different datasets. We prove that ACyLeR can achieve lower loss values and higher accuracy, from 0.5059 to 0.3915 in average MAE and from 0.5503 to 0.4350 in average MSE in the WSM dataset. Our strategy enhances model performance by dynamically adapting the learning rate according to the training progress, enabling the model to overcome local minima and produce superior overall results. We plan to incorporate additional attention mechanisms as the future direction of our research.

REFERENCES

[1] S. Geetha, C. Suwetha, A. Sangavi, and S. S. Nazrin, "Time series financial market forecasting based on machine learning," in *2024 2nd International Conference on Artificial Intelligence and Machine Learning Applications Theme: Healthcare and Internet of Things (AIMLA)*, 2024, pp. 1–5.

[2] A. T. Mohamed, H. H. Aly, and T. A. Little, "A comparative study of hourly wind speed and power forecasting using deep learning networks, weka time series, and arima algorithms for smart grid integration," in *2021 IEEE Electrical Power and Energy Conference (EPEC)*, 2021, pp. 273–278.

[3] C. Liao, J. Wang, B. Shan, J. Shang, T. Dong, and Y. He, "Near real-time detection and forecasting of within-field phenology of winter wheat and corn using sentinel-2 time-series data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 196, pp. 105–119, 2023.

[4] A. Ahmadi, A. Daccache, M. Sadegh, and R. L. Snyder, "Statistical and deep learning models for reference evapotranspiration time series forecasting: A comparison of accuracy, complexity, and data efficiency," *Computers and Electronics in Agriculture*, vol. 215, p. 108424, 2023.

[5] D. Yao and K. Yan, "Time series forecasting of stock market indices based on dlwr-lstm model," *Finance Research Letters*, p. 105821, 2024.

[6] P. Anuradha, V. Usha, K. N. Lakshman, P. L. Tejaswi, T. Anusha, and P. N. Sundari, "Comparison of time-series forecasting models based on prophets for predicting rainfall," in *2023 International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS)*, 2023, pp. 1542–1545. [Online]. Available: https://api.semanticscholar.org/CorpusID:263830644

[7] Y. Liu *et al.*, "itransformer: Inverted transformers are effective for time series forecasting," *ArXiv*, vol. abs/2310.06625, 2023, [Online]. Available:.

[8] S. Hanifi, A. Cammarono, and H. Zare-Behtash, "Advanced hyperparameter optimization of deep learning models for wind power prediction," *Renewable Energy*, vol. 221, p. 119700, 2024.

[9] B. Provencher, A. Badran, J. Kroll, and M. Marsh, "Hyperparameter tuning for deep learning semantic image segmentation of micro-computed tomography scanned fiber-reinforced composites," *Tomography of Materials and Structures*, p. 100032, 2024.

[10] J. Zhang, H. Liu, W. Bai, and X. Li, "A hybrid approach of wavelet transform, arima and lstm model for the share price index futures forecasting," *North American Journal of Economics and Finance*, vol. 69, p. 102022, 2024.

[11] Z. Sun, D. Song, Q. Peng, H. Li, and P. Li, "Improving long-term electricity time series forecasting in smart grid with a three-stage channel-temporal approach," *Journal of Cleaner Production*, vol. 468, p. 143051, 2024.

[12] L. Deng, X. Chang, and P. Wang, "Daily water demand prediction driven by multi-source data," *Procedia Computer Science*, vol. 208, pp. 128–135, 2022.

[13] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[14] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 22 419–22 430.

[15] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting," in *Proceedings of the 39th International Conference on Machine Learning*, vol. 162, 2022, pp. 27 268–27 286. [Online]. Available: https://proceedings.mlr.press/v162/zhou22g.html

[16] Y. Yu, R. Ma, and Z. Ma, "Robformer: A robust decomposition transformer for long-term time series forecasting," *Pattern Recognition*, p. 110552, 2024.

[17] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," *arXiv Preprints*, 2022, arXiv2211.14730.

[18] Y. Yang, Y. Zheng, R. Wang, K. Chen, C. Li, and Q. Guo, "Short-term prediction of photovoltaic power based on meteorological characteristics and deep learning in guizhou mountainous areas," in *2024 9th Asia Conference on Power and Electrical Engineering (ACPEE)*, 2024, pp. 449–455.

[19] A. Jha, O. Dorkar, A. Biswas, and A. Emadi, "itransformer network based approach for accurate remaining useful life prediction in lithium-ion batteries," in *2024 IEEE Transportation Electrification Conference and Expo (ITEC)*, 2024, pp. 1–8.

[20] Y. Yao, C. Gu, S. Shan, and K. Zhang, "Solar irradiance prediction with multi-meteorological variables based on itransformer," in *2024 IEEE 13th Data Driven Control and Learning Systems Conference (DDCLS)*, 2024, pp. 1579–1584.

[21] X. Zhang *et al.*, "A dynamic context-aware approach for vessel trajectory prediction based on multi-stage deep learning," *IEEE Transactions on Intelligent Vehicles*, pp. 1–16, 2024.

[22] H. Ni *et al.*, "Time series modeling for heart rate prediction: From arima to transformers," *arXiv Preprints*, 2024, arXiv2406.12199.

[23] J. Zhang, C. Tan, Z. Cai, L. Zhu, Y. Feng, and S. Liang, "Cellular traffic forecasting based on inverted transformer for mobile perception dual-level base station sleep control," *Ad Hoc Networks*, vol. 161, p. 103505, 2024.

[24] R. Swathika, S. M. D. Kumar, N. N. Srinidhi, and B. R. Harshitha, "Harnessing learn rate schedule for adaptive deep learning in lorawan-iot localization," *IEEE Access*, vol. 12, pp. 72 034–72 050, 2024.

[25] H. Iiduka, "-approximation of adaptive leaning rate optimization algorithms for constrained nonconvex stochastic optimization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 10, pp. 8108–8115, 2023.