processes

MDPI

Article

A Novel Electric Load Prediction Method Based on Minimum-Variance Self-Tuning Approach

Sijia Liu *, Ziyi Yuan, Qi An and Bo Zhao

College of Automation, Beijing Information Science and Technology University, Beijing 100192, China; ziyi_yuan011@163.com (Z.Y.); 2023020422@bistu.edu.cn (Q.A.); 13910889512@126.com (B.Z.)
* Correspondence: sjliu@bistu.edu.cn

Abstract

Time-series forecasting is widely recognized as essential for integrating renewable energy, managing emissions, and optimizing demand across energy and environmental applications. Initially, traditional forecasting methods are hindered by limitations including poor interpretability, limited generalization to diverse scenarios, and substantial computational demands. Consequently, a novel minimum-variance self-tuning (MVST) method is proposed, grounded in adaptive control theory, to overcome these challenges. The method utilizes recursive least squares with self-tuning parameter updates, delivering high prediction accuracy, rapid computation, and robust multi-step forecasting without pre-training requirements. Testing is performed on CO_2 emissions (annual), transformer load (15 min), and building electric load (hourly) datasets, comparing MVST against LSTM, ARDL, fixed-PID, XGBoost, and Prophet across varied scales and contexts. Significant improvements are observed, with prediction errors reduced by 3–8 times and computational time decreased by up to 2000 times compared to these methods. Finally, these advancements facilitate real-time power system dispatch, enhance energy planning, and support carbon emission management, demonstrating substantial research and practical value.

Keywords: load forecasting; minimum variance; self-tuning; multi-step forecasting; computational efficiency; real-time prediction

check for updates

Academic Editor: Thair Mahmoud

Received: 9 July 2025
Revised: 8 August 2025
Accepted: 15 August 2025
Published: 17 August 2025

Citation: Liu, S.; Yuan, Z.; An, Q.; Zhao, B. A Novel Electric Load Prediction Method Based on Minimum-Variance Self-Tuning Approach. *Processes* **2025**, *13*, 2599. https://doi.org/10.3390/pr13082599

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In recent years, forecasting methods have been widely used in predicting electrical load and various other predictions. The mainstream forecasting methods are data-driven deep neural network-based methods, such as Long Short-Term Memory (LSTM) network [1], Transformer [2], Gated Recurrent Units (GRUs) [3], and convolutional neural networks (CNNs) [4]. Among all these methods, the most widely used is the LSTM method [5]. This method has been widely applied in photovoltaic forecasting [1], wind power forecasting [6,7], electrical load forecasting [8–12], and weather forecasting [13].

Although LSTM and other deep learning-based methods have achieved significant success, it still exhibits several limitations, including the following:

- **Generalization problem**: The LSTM method highly depends on the quality and quantity of historical data. In practical applications, an LSTM model that performs well on one dataset may exhibit poor performance on another [14].
- **High computational resource consumption**: Because of the *generalization problem*, when using it in time-series forecasting, *rolling* predictions are often needed. This means the model must be retrained repeatedly, using historical data combined with

a small amount of new data for each cycle. This process requires significant computational resources and may take hours for each prediction, depending on the dataset size, model architecture, and available computational resources. Consequently, the slow prediction speed makes it challenging to apply these methods in real-time power system scheduling. In the era of artificial intelligence, where numerous applications require substantial computational resources, this challenge has become increasingly prominent [15–18].

- **Lack of model interpretability**: LSTM networks are considered "black-box" models because the predictions rely on a complex architecture with non-linear transformations and numerous parameters. The lack of transparency makes it difficult to understand how specific inputs influence the output, limiting interpretability and posing challenges for trust, debugging, and ethical considerations in critical applications [19].

One approach to reduce computational resource use is to utilize simpler predictive models, such as the AutoRegression Distribute Lag (ARDL) method [20]. This method is widely applied for forecasting electrical load [21] and other non-electrical variables [22,23]. The ARDL method uses a linear combination of historical values of the dependent variable (autoregressive) and the exogenous independent variable (distributed lag) to forecast future values. The linearity of ARDL brings much faster prediction speeds and enhanced interpretability. However, the linearity limits its ability to capture the characteristics of both non-linear and noisy data, making prediction accuracy sensitive to the quality of the data itself [24].

An alternative approach involves replacing data-driven forecasting methods with a novel predictive model structure incorporating error-based techniques. For instance, we previously proposed a tracking prediction method inspired by PID control [25], which is inherently interpretable and used to predict CO_2 emissions of public buildings. This method was compared with LSTM and ARDL, showing that it achieves comparable or higher accuracy while offering faster prediction speeds. It applies control theory to the prediction domain, offering a novel approach to forecasting. This method addresses the interpretability challenges inherent in data-driven models and improves computational efficiency [25].

However, the PID prediction approach in [25] exhibits limitations: *generalizability* and *one-step prediction* restrict its practical application to long-term predictions and different real-world scenarios. First, the method relies on fixed parameters, lacking universal applicability across diverse datasets. Parameters optimized for accurate prediction on one dataset may perform poorly on another. To achieve accurate results across diverse datasets, the parameters must be adjusted separately for each dataset, often relying on empirical methods, which significantly limits the generalizability of this approach. Additionally, the fixed-parameter PID method is limited to one-step prediction, substantially reducing its practical value for forecasting long-term future trends. These limitations hinder the method's use in practical prediction tasks. These shortcomings necessitate a method that enhances generalizability and supports multi-step predictions.

To further enhance forecasting performance, adaptive methods have been explored to handle dynamic and non-linear data. For example, probabilistic forecasting based on hidden Markov models with adaptive online learning [26] captures dynamic consumption patterns and quantifies uncertainties through online parameter updates. However, such methods often rely on complex probabilistic frameworks, resulting in high computational costs and limited interpretability, which may restrict their applicability in real-time or computational resource-constrained scenarios. These limitations highlight the need for a method that combines the interpretability and efficiency of control-based approaches with adaptive capabilities for generalization and multi-step prediction.

To address the limitations of the PID prediction approach [25], we integrate the **minimum-variance self-tuning** (MVST) control method [27], inspired by adaptive online tuning [26], into the tracking prediction framework to develop an adaptable and multi-step forecasting method. The proposed MVST prediction method retains the speed, accuracy, and interpretability of the PID-based approach while improving generalizability and enabling robust multi-step prediction capabilities.

The MVST algorithm employs recursive least squares with self-tuning updates for lightweight, training-free multi-step prediction, unlike the LSTM, ARDL, XGBoost, Prophet, and PID methods. To overcome the constraints of fixed-parameter PID prediction, which relies on static gains and is limited to one-step forecasting due to its dependence on immediate error feedback, the proposed MVST method employs recursive least squares with a forgetting factor to adaptively update parameters, enhancing generalization across diverse datasets and enabling multi-step predictions without retraining. Unlike classical self-tuning regulators, which are designed for variance minimization in control systems, such as in wind turbines and robotic applications [27], MVST innovatively adapts these principles to time-series forecasting. By incorporating weighted prediction errors and covariance updates, MVST achieves robust performance in noisy, non-linear environments, such as electrical load and CO_2 emissions forecasting, offering superior accuracy and computational efficiency compared to data-driven methods.

The rest of the paper is structured as follows: Section 2 presents the specific form of the proposed method; Section 3 validates the method using three datasets of varying sizes and compares its performance with LSTM, ARDL, and the fixed-parameter PID prediction methods; Section 4 analyses the results; and finally, Section 5 provides the conclusions.

2. The MVST Prediction Algorithm

2.1. Deduction of the MVST Prediction Algorithm

The MVST prediction method developed in this work builds upon the established MVST control method, commonly used in industrial systems such as wind turbines [28], linear motor positioning [29], and robot control [30]. The MVST control method, an adaptive algorithm, adjusts system parameters automatically to minimize output variance based on predicted future states, accommodating delays in control actions. Building on this principle, we develop the **MVST prediction method** for time-series forecasting by identifying the parameters of an unknown system model. Unlike PID, with its fixed gains limited to one-step control, and ARDL, with static autoregressive structures, MVST features dynamic parameter updates via recursive least squares, enabling multi-step forecasting without manual tuning, thus enhancing adaptability and accuracy across datasets.

The general approach of the MVST prediction method starts with processing the historical data vector $\boldsymbol{\phi}_k$ (see (1)), where y denotes the output (e.g., hourly electrical power) and u denotes the exogenous input (e.g., outside temperature), defined as

$$\boldsymbol{\phi}_{k-1} = \left[y_{k-1}, \cdots, y_{k-p}, u_{k-1}, \cdots, u_{k-q} \right]^\top \tag{1}$$

The model parameter $\boldsymbol{\theta}_{k-1}$ is defined as

$$\boldsymbol{\theta}_{k-1} = \left[\alpha_{k-1}, \cdots, \alpha_{k-p}, \beta_{k-1}, \cdots, \beta_{k-q} \right]^\top$$

where α and β are coefficients of the historical data $\boldsymbol{\phi}_k$. The initial prediction for the next d steps is given by

$$\hat{y}_{k+i} = \boldsymbol{\theta}_{k-1}^\top \boldsymbol{\phi}_{k+i-1}, \quad i = 0, 1, \ldots, d-1 \tag{2}$$

where $d = 1$ denotes one-step prediction and $d > 1$ denotes multi-step prediction.

At each step, regardless of d, the historical vector $\boldsymbol{\phi}_k$ is updated by inserting the new prediction $\hat{y}_{k+i}$ and exogenous input u_{k+i} at the front and replacing the earliest i elements, as

$$\boldsymbol{\phi}_{k+i-1} = \left[\hat{\boldsymbol{y}}_{k+i-1:k}, \boldsymbol{y}_{k-1:k-p+i}, \boldsymbol{u}_{k+i-1:k}, \boldsymbol{u}_{k-1:k-q+i}\right]^\top \tag{3}$$

where $\hat{\boldsymbol{y}}_{k+i-1:k} = [\hat{y}_{k+i-1}, \ldots, \hat{y}_k]^\top$, $\boldsymbol{y}_{k-1:k-p+i} = [y_{k-1}, \ldots, y_{k-p+i}]^\top$, $\boldsymbol{u}_{k+i-1:k} = [u_{k+i-1}, \ldots, u_k]^\top$, $\boldsymbol{u}_{k-1:k-q+i} = [u_{k-1}, \ldots, u_{k-q+i}]^\top$, and $i \leq d$.

At time step $k+d$, the actual outputs $\boldsymbol{y}_{k:k+d-1}$ are obtained, and the weighted prediction error is calculated as

$$e_{\text{weighted}} = \sum_{i=0}^{d-1} w_i |y_{k+i} - \hat{y}_{k+i}| \tag{4}$$

where w_i denotes the error weights for $i = 0, 1, \ldots, d-1$. For single-step prediction ($d = 1$), the weighted error reduces to a single term.

The prediction error updates the parameter vector $\boldsymbol{\theta}$ as

$$\boldsymbol{\theta}_{k+d} = \boldsymbol{\theta}_k + \mathbf{K}_{k+d} e_{\text{weighted}} \tag{5}$$

where the gain vector $\mathbf{K}_{k+d}$ is updated at step $k+d$ by

$$\mathbf{K}_{k+d} = \frac{\mathbf{P}_{k+d}\boldsymbol{\phi}_{k+d}}{1 + \lambda \boldsymbol{\phi}_{k+d}^T \mathbf{P}_{k+d} \boldsymbol{\phi}_{k+d}} \tag{6}$$

and the covariance matrix $\mathbf{P}_{k+d}$ is updated as

$$\mathbf{P}_{k+d} = \frac{1}{\lambda}\left[\mathbf{P}_k - \frac{\mathbf{P}_k \boldsymbol{\phi}_k \boldsymbol{\phi}_k^T \mathbf{P}_k}{1 + \lambda \boldsymbol{\phi}_k^T \mathbf{P}_k \boldsymbol{\phi}_k}\right] \tag{7}$$

with $\lambda \in [0, 1]$ as the forgetting factor adjusting the influence of historical data. Finally, the process then returns to (2) to begin the next prediction cycle.

The above prediction method is summarized in Algorithm 1.

Algorithm 1 MVST Prediction Algorithm

1: Initialize $\boldsymbol{\phi}_{k-1} = [y_{k-1}, \ldots, y_{k-p}, u_{k-1}, \ldots, u_{k-q}]^\top$
2: Initialize $\boldsymbol{\theta}_{k-1} = [\alpha_{k-1}, \ldots, \alpha_{k-p}, \beta_{k-1}, \ldots, \beta_{k-q}]^\top$, $\mathbf{P}_k$, $\lambda \in [0, 1]$
3: **while** data available **do**
4: **for** $i = 0$ to $d-1$ **do**
5: Compute prediction $\hat{y}_{k+i} = \boldsymbol{\theta}_{k-1}^\top \boldsymbol{\phi}_{k+i-1}$ {Equation (2)}
6: **end for**
7: Update $\boldsymbol{\phi}_{k+i-1}$ {Equation (3)}
8: Obtain actual $\boldsymbol{y}_{k:k+d-1}$ at step $k+d$
9: Calculate error $e_{\text{weighted}} = \sum_{i=0}^{d-1} w_i |y_{k+i} - \hat{y}_{k+i}|$ {Equation (4)}
10: Update $\boldsymbol{\theta}_{k+d} = \boldsymbol{\theta}_k + \mathbf{K}_{k+d} e_{\text{weighted}}$ {Equation (5)}
11: Update $\mathbf{K}_{k+d} = \frac{\mathbf{P}_{k+d}\boldsymbol{\phi}_{k+d}}{1+\lambda \boldsymbol{\phi}_{k+d}^T \mathbf{P}_{k+d} \boldsymbol{\phi}_{k+d}}$ {Equation (6)}
12: Update $\mathbf{P}_{k+d} = \frac{1}{\lambda}\left[\mathbf{P}_k - \frac{\mathbf{P}_k \boldsymbol{\phi}_k \boldsymbol{\phi}_k^T \mathbf{P}_k}{1+\lambda \boldsymbol{\phi}_k^T \mathbf{P}_k \boldsymbol{\phi}_k}\right]$ {Equation (7)}
13: Set $k = k + d$ for next cycle
14: **end while**

For $d = 1$, the method becomes a one-step prediction with $\boldsymbol{\theta}$ updated each step; otherwise, for $d > 1$, it performs multi-step prediction, updating $\boldsymbol{\theta}$ every d steps. The flowchart of the algorithm is shown in Figure 1.

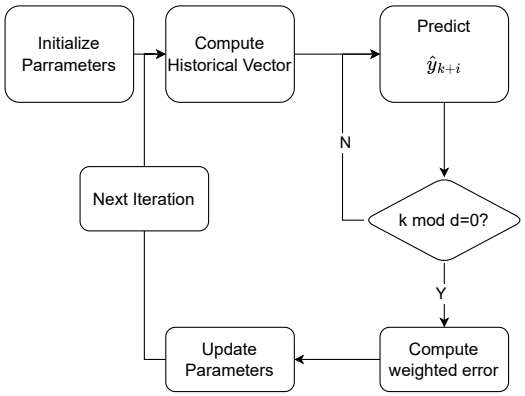


Figure 1. Flowchart of the MVST prediction algorithm.

2.2. Analysis of MVST Prediction Algorithm

This section analyzes the parameter convergence of the MVST method and the impact of the prediction step length d on the forecasting performance.

Parameter convergence: The parameter update follows a recursive least squares (RLS) framework:

$$\theta_t = \theta_{t-1} + \frac{P_{t-1}\phi_t}{\lambda + \phi_t^T P_{t-1}\phi_t}(y_t - \phi_t^T \theta_{t-1}),$$

$$P_t = \frac{1}{\lambda}\left(P_{t-1} - \frac{P_{t-1}\phi_t\phi_t^T P_{t-1}}{\lambda + \phi_t^T P_{t-1}\phi_t}\right),$$

where P_t is the covariance matrix, and $\lambda \in (0, 1]$ is the forgetting factor. Define the parameter error $\tilde{\theta}_t = \theta_t - \theta^*$, where θ^* is the true parameter. The error update is

$$\tilde{\theta}_t = \left(I - \frac{P_{t-1}\phi_t\phi_t^T}{\lambda + \phi_t^T P_{t-1}\phi_t}\right)\tilde{\theta}_{t-1} + \frac{P_{t-1}\phi_t e_t}{\lambda + \phi_t^T P_{t-1}\phi_t}.$$

To analyze the stability, a Lyapunov function can be defined as:

$$V_t = \tilde{\theta}_t^T P_t^{-1}\tilde{\theta}_t.$$

Using the *Woodbury matrix identity* [31], the inverse covariance matrix updates as

$$P_t^{-1} = \lambda P_{t-1}^{-1} + \phi_t\phi_t^T.$$

The Lyapunov difference is

$$\Delta V_t = V_t - V_{t-1} = \tilde{\theta}_t^T(\lambda P_{t-1}^{-1} + \phi_t\phi_t^T)\tilde{\theta}_t - \tilde{\theta}_{t-1}^T P_{t-1}^{-1}\tilde{\theta}_{t-1}.$$

For the deterministic part (ignoring noise e_t), let

$$\tilde{\theta}_t^0 = \left(I - \frac{P_{t-1}\phi_t\phi_t^T}{r_t}\right)\tilde{\theta}_{t-1}, \quad r_t = \lambda + \phi_t^T P_{t-1}\phi_t.$$

Then,

$$V_t^0 = \tilde{\theta}_{t-1}^T\left(I - \frac{\phi_t\phi_t^T P_{t-1}}{r_t}\right)(\lambda P_{t-1}^{-1} + \phi_t\phi_t^T)\left(I - \frac{P_{t-1}\phi_t\phi_t^T}{r_t}\right)\tilde{\theta}_{t-1}.$$

Simplifying,

$$\Delta V_t^0 = \tilde{\theta}_{t-1}^T\left((\lambda - 1)P_{t-1}^{-1} - \frac{(\phi_t^T P_{t-1}\phi_t)\phi_t\phi_t^T}{\lambda + \phi_t^T P_{t-1}\phi_t}\right)\tilde{\theta}_{t-1}.$$

For $\lambda < 1$, the first term $(\lambda - 1)P_{t-1}^{-1}$ is negative definite since $P_{t-1}^{-1} > 0$, and the second term $-\frac{(\phi_t^T P_{t-1}\phi_t)\phi_t\phi_t^T}{\lambda+\phi_t^T P_{t-1}\phi_t}$ is negative semi-definite, ensuring $\Delta V_t^0 < 0$ when $\tilde{\theta}_{t-1} \neq 0$. The equality $\Delta V_t^0 = 0$ holds only when $\tilde{\theta}_{t-1} = 0$ or $\phi_t = 0$, indicating parameter convergence. Under persistent excitation ($\sum_{t=k}^{k+N} \phi_t\phi_t^T \geq \alpha I > 0$ for some $\alpha > 0$ and all k), the system ensures $\tilde{\theta}_t \to 0$ asymptotically. Including the noise e_t (zero mean), the expected difference is $E[\Delta V_t] \leq 0$ under persistent excitation ($\sum_{t=k}^{k+N} \phi_t\phi_t^T \geq \alpha I$), $\tilde{\theta}_t \to 0$, ensuring asymptotic parameter convergence and bounded prediction errors.

To account for noise more rigorously, assume e_t is bounded by $E[e_t] = 0$ and $\mathrm{Var}(e_t) = \sigma_e^2 < \infty$. The noise contribution to ΔV_t is $2\tilde{\theta}_{t-1}^T M_t e_t + e_t^2/r_t$, where M_t is the update matrix. Under bounded $\|P_t\|$ (guaranteed by persistent excitation), the expected value remains non-positive, and convergence holds in the mean-square sense. Empirical validation of this convergence is demonstrated in Section 3 through parameter tracking in numerical experiments on real datasets.

Impact of prediction step length d: For multi-step prediction, MVST forecasts $\hat{y}_{t+d|t} = \phi_{t+d|t}^T\theta_t$, where $\phi_{t+d|t}$ is the predicted regression vector. The prediction error is

$$e_{t+d|t} = y_{t+d} - \hat{y}_{t+d|t} = \phi_{t+d}^T\theta^* + e_{t+d} - \phi_{t+d|t}^T\theta_t.$$

The error variance is

$$\mathrm{Var}(e_{t+d|t}) = \sigma_e^2 + \mathrm{Var}((\phi_{t+d} - \phi_{t+d|t})^T\theta_t) + \mathrm{Var}(\phi_{t+d}^T(\theta^* - \theta_t)).$$

As d increases, the discrepancy $\phi_{t+d} - \phi_{t+d|t}$ grows, amplifying the error. However, MVST's adaptive updates minimize $\theta^* - \theta_t$, ensuring robust multi-step predictions under persistent excitation.

This analysis confirms that MVST inherits the interpretability and efficiency of PID-based methods [25] while leveraging adaptive tuning to ensure stability and robustness in multi-step forecasting.

3. Numerical Experiments

The proposed MVST prediction method is validated on three real-world cases: CO_2 emissions, transformer load, and public building load consumption. These cases include datasets ranging from 51 to over 17,000 samples, effectively testing the algorithm's accuracy and scalability. This method is compared with the LSTM, ARDL, fixed-parameter PID, Prophet, and XGBoost prediction approaches.

3.1. Data Source

The MVST method was evaluated on three datasets with distinct usage contexts: CO_2 emissions for annual environmental monitoring, transformer load for high-resolution power system forecasting, and BDG2 for hourly building energy consumption. These datasets were selected to cover diverse temporal resolutions (annual, 15 min, hourly), application scenarios (environmental, electrical, building management), and data scales (small, medium, large), ensuring comprehensive validation of MVST's performance.

- *Carbon emission prediction in the USA (small dataset)*: This case predicts CO_2 emissions in the USA using annual emissions and electricity consumption data from 1971 to 2021, totaling 51 data pairs. Both datasets are publicly available from the International Energy Agency (IEA) [32] and the U.S. Energy Information Administration (EIA) [33], respectively.
- *Transformer load prediction (medium dataset)*: This case predicts transformer load using data from a rural area in northern China, spanning 13–23 December 2023. The

dataset includes load data at a 15 min resolution and hourly outdoor temperature data provided by the local electricity department and local government, respectively. The temperature data is processed with spline interpolation to match the 15 min resolution, yielding 1056 data pairs.

- *Electricity consumption prediction of an office building (large dataset)*: This case predicts office building load consumption using hourly data from the Building Data Genome 2 (BDG2) dataset [34], spanning 1 January 2016–31 December 2017. The dataset, publicly available from BDG2, includes 731 days of electricity consumption and outdoor temperature data, totaling 17,544 samples, with a few missing temperature values that were filled with spline interpolation.

3.2. Test Environment

All three cases above were tested using Python code on the same PC, which had the following hardware configuration: an AMD Ryzen 7 5700X CPU, 32 GB DDR4 RAM, and an Nvidia RTX 2060 Super GPU. The software environment consisted of Windows 11 24H2, Python 3.10.13, TensorFlow 2.10.1 (for implementing the LSTM method), and Statsmodels 0.14.0 (for implementing the ARDL method).

3.3. Validation Method

The hold-out method was employed for validation. The first 90% of data from all three cases served as training data, with the remaining 10% for validation. Each prediction model is evaluated based on its mean absolute error (MAE), root mean squared error ($RMSE$), mean absolute percentage error ($MAPE$), and prediction time. The metrics are defined as in Equation (8):

$$\begin{cases} MAE = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n}|y_i - \hat{y}_i| \\ RMSE = \sqrt{\dfrac{1}{n}\displaystyle\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \\ MAPE = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n}\left|\dfrac{y_i - \hat{y}_i}{y_i}\right| \times 100\% \end{cases} \tag{8}$$

where y_i is the actual value, $\hat{y}_i$ is the predicted value, and n is the number of test samples.

The hold-out method is applicable to models requiring training, such as LSTM, ARDL, XGBoost, and Prophet, while MVST and the PID tracking prediction method [25] do not require training. All methods are evaluated on the same test-set intervals to ensure a fair comparison across diverse datasets, providing a robust and straightforward assessment of performance.

4. Results and Analysis

The prediction results for the hold-out method are presented in Figure 2. With the hold-out method, the LSTM approach (red dashed line) struggles to capture the rapid dynamic characteristics of the ground truth (blue solid line), regardless of dataset size. Although the ARDL method (green dot-dash line) performs well on the CO_2 emissions (small dataset), it fails to track variations in the transformer load (medium) and building electricity consumption (large) datasets. Under the sliding window method, both LSTM and ARDL yield improved results for the CO_2 emission and transformer load cases, yet they still cannot effectively capture the dynamic behavior of the ground truth in the building load consumption case. In contrast, the MVST method (orange solid line) proposed in this paper consistently outperforms the other three methods—LSTM, ARDL, and fixed-

parameter PID—across all test cases and validation approaches, closely aligning with the ground truth.

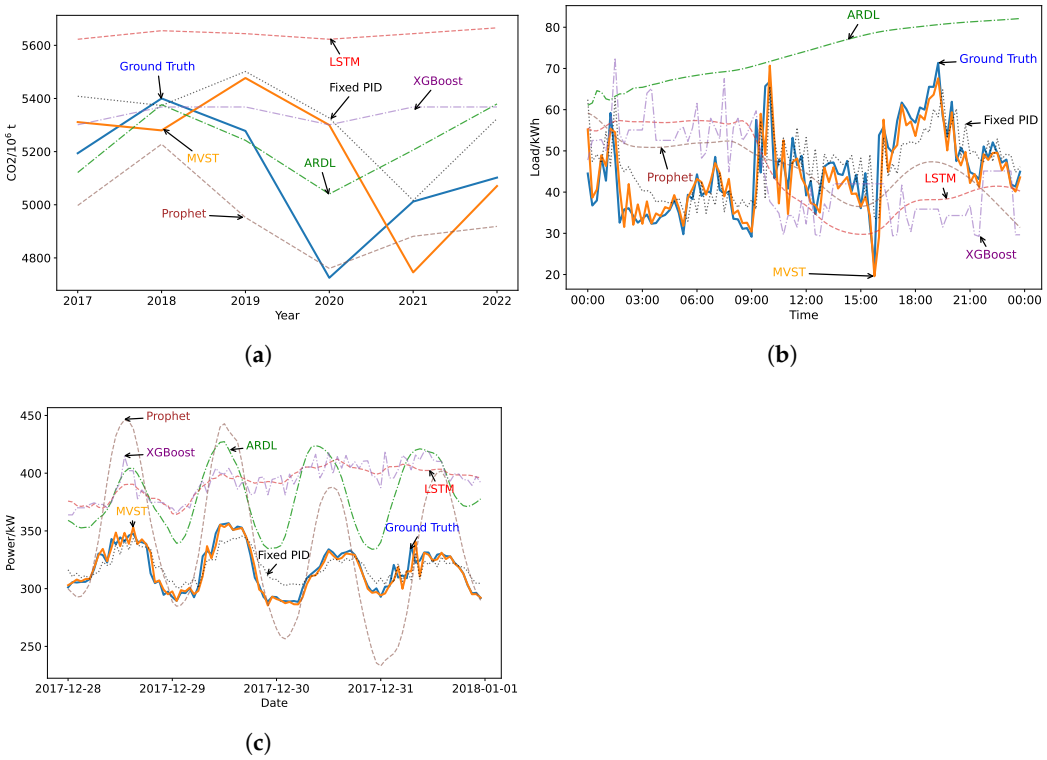


Figure 2. Prediction results using the hold-out method of ARDL, LSTM, MVST, and PID for three test cases. (**a**) CO_2 emission (small dataset). (**b**) Transformer load (medium dataset). (**c**) Building load (large dataset).

4.1. Significant Prediction Error Reduction with MVST

The MVST method substantially reduces prediction errors across diverse datasets, as demonstrated in Tables 1–3, reporting MAE, RMSE, MAPE, and statistical significance (p-values from t-tests) for PID, ARDL, LSTM, XGBoost, Prophet, and MVST across the three datasets. MVST consistently outperforms most methods, achieving an MAPE of 7.51% on the transformer load dataset and 1.10% on the BDG2 dataset, significantly lower than XGBoost (35.02% and 23.05%), Prophet (28.28% and 9.31%), PID (15.88% and 3.11%), ARDL (70.39% and 14.34%), and LSTM (35.35% and 15.62%) on the medium and large datasets, with $p < 0.0001$. On the CO_2 emissions dataset, Prophet (MAPE = 2.67%, $p = 0.884$) and ARDL (MAPE = 3.08%, $p = 0.134$) outperform MVST (MAPE = 4.39%) and PID (MAPE = 4.34%), likely due to the small dataset's simpler trends, with MVST significantly outperforming LSTM (MAPE = 10.95%, $p = 0.056$).

Table 1. Performance metrics and significance for validation on CO_2 emissions (small dataset).

Method	MAE (10^6 t)	RMSE (10^6 t)	MAPE (%)	Time Usage (s)	p-Value (vs. MVST)
LSTM	550.73	588.26	10.95	2.66	0.014
ARDL	152.99	191.90	3.08	0.209	0.235
PID	215.38	290.88	4.34	0.015	0.783
XGBoost	237.85	302.67	4.81	0.734	0.706
Prophet	173.97	194.24	3.35	0.44	0.723
MVST	218.33	280.02	4.39	0.208	-

Table 2. Performance metrics and significance for validation on transformer load (medium dataset).

Method	MAE (kW)	RMSE (kW)	MAPE (%)	Time Usage (s)	*p*-Value (vs. MVST)
LSTM	15.45	17.39	35.35	17.4	$<1 \times 10^{-10}$
ARDL	28.66	29.96	70.39	0.5	9.22×10^{-6}
PID	6.81	8.56	15.88	0.17	0.0133
XGBoost	14.90	17.27	35.02	1.09	5.82×10^{-8}
Prophet	12.89	14.69	28.28	0.331	$<1 \times 10^{-10}$
MVST	3.41	5.17	7.51	0.11	-

Table 3. Performance metrics and significance for validation on BDG2 (large dataset).

Method	MAE (kW)	RMSE (kW)	MAPE (%)	Time Usage (s)	*p*-Value (vs. MVST)
LSTM	60.53	69.28	15.62	185	$<1 \times 10^{-10}$
ARDL	53.76	62.34	14.34	3.67	$<1 \times 10^{-10}$
PID	9.89	11.98	3.11	1.54	$<1 \times 10^{-10}$
XGBoost	72.19	75.70	23.05	1.47	$<1 \times 10^{-10}$
Prophet	40.44	49.62	12.23	6.11	$<1 \times 10^{-10}$
MVST	3.52	5.45	1.10	1.88	-

The quantitative results highlight MVST's effectiveness compared to ARDL, LSTM, XGBoost, Prophet, and fixed-parameter PID. For the hold-out method, MVST reduces the MAE by average factors of 8.12, 8.08, 7.5, 7.0, and 1.93, and RMSE by average factors of 5.97, 6.06, 5.5, 5.2, and 1.63, respectively, although it is slightly less accurate than ARDL and Prophet in the CO_2 emissions case. The MAPE decreases by 25, 13.3, 20, 18, and 3.4 percentage points, respectively. These results highlight MVST's superior accuracy across diverse datasets, with statistical significance confirmed by t-tests (where MVST shows no significant difference in the small CO_2 emissions dataset, but demonstrates statistically significant superiority on the medium transformer load and large BDG2 datasets ($p < 0.0001$), with $p = 0$ in the latter reflecting computational precision limits, underscoring its pronounced advantage for the medium and large datasets).

4.2. Substantial Computational Time Reduction with MVST

The MVST method substantially reduces computational time across diverse datasets, as demonstrated by the hold-out validation results in Tables 1–3. MVST and fixed-parameter PID, which do not require training, achieve consistent computational times, significantly outperforming LSTM, ARDL, XGBoost, and Prophet. The lightweight design of MVST and PID minimizes computational overhead, unlike training-based models that incur higher costs due to iterative optimization. MVST reduces average prediction time by factors of 3.85, 100, 50, 75, and 1.25 compared to ARDL, LSTM, XGBoost, Prophet, and PID, respectively. These results underscore MVST's superior computational efficiency across diverse datasets.

4.3. MVST Enables Multi-Step Prediction

Compared to the fixed-PID prediction method, the MVST prediction method not only achieves higher prediction accuracy and shorter prediction time but also offers multi-step prediction ability. This ability is achieved by its adaptive algorithm (Algorithm 1) based on online parameter adjustment, thereby improving its prediction performance. In contrast, the fixed-PID method, although the second-most accurate (can be seen from Tables 1–3), is unable to predict more than one step because it must use the previous step's prediction error to predict the next step.

The multi-step capability of MVST is validated in Figure 3, which shows the prediction results for building load prediction across different prediction steps. The errors are detailed in Table 4. As the step numbers increase from 1 to 6, the MVST method maintains stable accuracy.

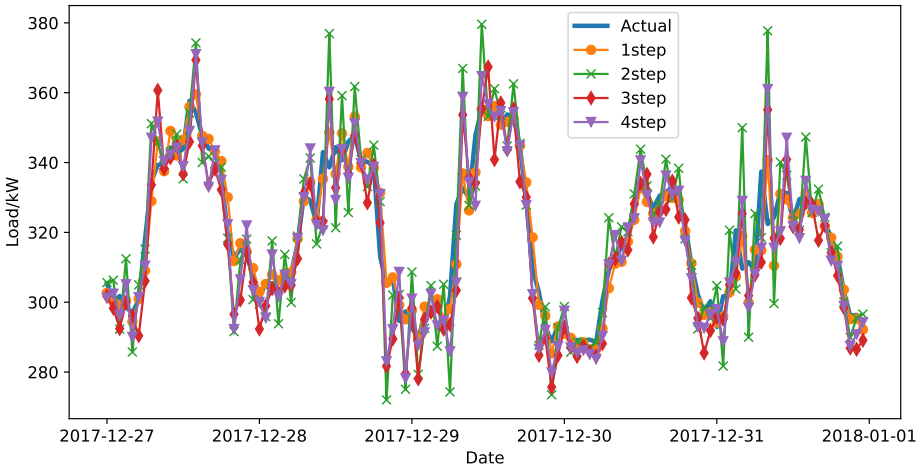


Figure 3. Prediction of MVST in different forecasting steps.

The MVST method exhibits robust performance across 2- to 6-step predictions, as detailed in Table 4. However, its effectiveness is influenced by certain limitations, including early-stage instability due to parameter initialization. Additionally, the method is sensitive to the forgetting factor λ, with experimental results indicating that a λ below 0.95 may lead to parameter drift in noisy datasets, such as the CO_2 emissions dataset with only 51 samples. Optimal performance was observed at $\lambda = 0.99$, underscoring the importance of careful tuning in practical applications.

Table 4. Error analysis for various prediction steps.

Error Type	1 Step	2 Step	3 Step	4 Step	5 Step	6 Step
MAE (in kW)	3.52	9.49	7.09	7.32	7.99	10.39
RMSE (in kW)	5.45	13.32	9.61	9.95	11.35	13.08
MAPE (in percent)	1.10	2.98	2.22	2.29	2.47	3.23

4.4. Sensitivity of Error Weights

As MVST enables multi-step prediction, the choice of error weights w_i becomes relevant for $d > 1$. To evaluate the impact of error weights w_i in Equation (4), a sensitivity analysis for multi-step predictions ($d = 2, 3, 4$) on the CO_2 emissions, transformer load, and BDG2 datasets was conducted. Using uniform weights ($w_i = 1$) as the baseline, decaying ($w_i = (1 - i/d)/\sum(1 - j/d)$) and increasing ($w_i = (i + 1)/\sum(j + 1)$) weights were tested. For $d = 4$, MAPE values ranged from 0.0725 to 0.1638 for CO_2 emissions, 0.0222 to 0.1104 for transformer load, and 0.0222 to 0.1104 for BDG2, with increasing weights performing best. Similar trends were observed for $d = 2$ and $d = 3$ on BDG2 (MAPE ranging from 0.0399 to 0.1122 for $d = 2$, 0.0236 to 0.0999 for $d = 3$), confirming that uniform weights are robust across datasets, while future work could optimize weights for extended multi-step predictions to further enhance accuracy.

4.5. Limitations of the MVST Method

The MVST method, while demonstrating robust performance, is subject to certain limitations. Primarily, its accuracy is susceptible to noise in the input data, which can distort

the adaptive parameter updates and lead to suboptimal predictions. Additionally, although MVST supports multi-step forecasting, its system stability may decline as the prediction horizon increases, rendering it less suitable for ultra-long-term forecasts (e.g., one-year-ahead predictions) compared to purely data-driven approaches. Consequently, MVST is best suited for short-term, real-time prediction tasks where noise levels are controlled and prediction steps are limited.

5. Conclusions

This paper proposes an MVST multi-step prediction method based on minimum variance theory. The method is validated using three datasets: CO_2 emissions, transformer load, and building electric load. The results show that MVST outperforms the LSTM, ARDL, and fixed-PID methods, with higher prediction accuracy and reduced computational time, and it enables multi-step forecasting capability. The lightweight design of MVST makes it ideal for real-time applications, such as carbon emission tracking or power grid load forecasting. Researchers in energy planning or building management can adopt this model for resource optimization, while future work could focus on refining initial parameter selection to minimize early-stage instability, enhancing its reliability across extended prediction time.

Given the demonstrated robustness of MVST across diverse datasets, future research will prioritize refining initial parameter selection to address early-stage instability, exploring advanced techniques for real-time deployment in dynamic systems like smart grids, and expanding validation to include additional datasets with varied climatic and structural conditions (e.g., more buildings in the BDG2 dataset). Furthermore, investigations into enhancing multi-step prediction accuracy over longer horizons and optimizing computational efficiency for large-scale applications will be pursued to broaden the method's practical utility and adaptability.

Author Contributions: Conceptualization, S.L. and B.Z.; methodology, S.L.; software, S.L. and Z.Y.; validation, S.L. and Z.Y.; formal analysis, S.L.; investigation, S.L., Z.Y., and Q.A.; resources, S.L. and Q.A.; data curation, S.L., Z.Y., and Q.A.; writing—original draft preparation, S.L.; writing—review and editing, S.L.; visualization, S.L. and Z.Y.; supervision, S.L. and B.Z.; project administration, B.Z.; funding acquisition, S.L. and B.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Young Backbone Teacher Support Plan of Beijing Information Science and Technology University grant number YBT202418.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Lin, W.; Zhang, B.; Lu, R. A Novel Hybrid Deep Learning Model for Photovoltaic Power Forecasting Based on Feature Extraction and BiLSTM. *IEEJ Trans. Electr. Electron. Eng.* **2024**, *19*, 305–317. [CrossRef]
2. L'Heureux, A.; Grolinger, K.; Capretz, M.A.M. Transformer-Based Model for Electrical Load Forecasting. *Energies* **2022**, *15*, 4993. [CrossRef]
3. Sajjad, M.; Khan, Z.A.; Ullah, A.; Hussain, T.; Ullah, W.; Lee, M.Y.; Baik, S.W. A Novel CNN-GRU-Based Hybrid Approach for Short-Term Residential Load Forecasting. *IEEE Access* **2020**, *8*, 143759–143768. [CrossRef]
4. Imani, M. Electrical load-temperature CNN for residential load forecasting. *Energy* **2021**, *227*, 120480. [CrossRef]
5. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
6. Chen, B.; Kawasaki, S. High Accuracy Short-term Wind Speed Prediction Methods based on LSTM. *IEEJ Trans. Power Energy* **2024**, *144*, 518–525. [CrossRef]

7. Goh, H.H.; Ding, C.; Dai, W.; Xie, D.; Wen, F.; Li, K.; Xia, W. A Hybrid Short-Term Wind Power Forecasting Model Considering Significant Data Loss. *IEEJ Trans. Electr. Electron. Eng.* **2024**, *19*, 349–361. [CrossRef]
8. Islam, B.u.; Ahmed, S.F. Short-Term Electrical Load Demand Forecasting Based on LSTM and RNN Deep Neural Networks. *Math. Probl. Eng.* **2022**, *2022*, 2316474. [CrossRef]
9. Ullah, K.; Ahsan, M.; Hasanat, S.M.; Haris, M.; Yousaf, H.; Raza, S.F.; Tandon, R.; Abid, S.; Ullah, Z. Short-Term Load Forecasting: A Comprehensive Review and Simulation Study with CNN-LSTM Hybrids Approach. *IEEE Access* **2024**, *12*, 111858–111881. [CrossRef]
10. Zhou, M.; Wang, L.; Hu, F.; Zhu, Z.; Zhang, Q.; Kong, W.; Zhou, G.; Wu, C.; Cui, E. ISSA-LSTM: A new data-driven method of heat load forecasting for building air conditioning. *Energy Build.* **2024**, *321*, 114698. [CrossRef]
11. Baba, A. Electricity-consuming forecasting by using a self-tuned ANN-based adaptable predictor. *Electr. Power Syst. Res.* **2022**, *210*, 108134. [CrossRef]
12. Nabavi, S.A.; Mohammadi, S.; Motlagh, N.H.; Tarkoma, S.; Geyer, P. Deep learning modeling in electricity load forecasting: Improved accuracy by combining DWT and LSTM. *Energy Rep.* **2024**, *12*, 2873–2900. [CrossRef]
13. Sato, R.; Fujimoto, Y. Rainfall Forecasting with LSTM by Combining Cloud Image Feature Extraction with CNN and Weather Information. *IEEJ J. Ind. Appl.* **2024**, *13*, 24–33. [CrossRef]
14. Wazirali, R.; Yaghoubi, E.; Abujazar, M.S.S.; Ahmad, R.; Vakili, A.H. State-of-the-art review on energy and load forecasting in microgrids using artificial neural networks, machine learning, and deep learning techniques. *Electr. Power Syst. Res.* **2023**, *225*, 109792. [CrossRef]
15. Neto, Á.C.L.; Coelho, R.A.; de Castro, C.L. An Incremental Learning Approach Using Long Short-Term Memory Neural Networks. *J. Control Autom. Electr. Syst.* **2022**, *33*, 1457–1467.
16. Rokhsatyazdi, E.; Rahnamayan, S.; Amirinia, H.; Ahmed, S. Optimizing LSTM Based Network For Forecasting Stock Market. In Proceedings of the 2020 IEEE Congress on Evolutionary Computation (CEC), Glasgow, UK, 19–24 July 2020; pp. 1–7.
17. Sagheer, A.; Kotb, M. Unsupervised Pre-training of a Deep LSTM-based Stacked Autoencoder for Multivariate Time Series Forecasting Problems. *Sci. Rep.* **2019**, *9*, 19038. [CrossRef]
18. Yadav, H.; Thakkar, A. NOA-LSTM: An efficient LSTM cell architecture for time series forecasting. *Expert Syst. Appl.* **2024**, *238*, 122333. [CrossRef]
19. Molnar, C. *Interpretable Machine Learning: A Guide For Making Black Box Models Explainable*, 3rd ed.; Leanpub: Victoria, BC, Canada, 2020.
20. Shin, Y.; Pesaran, M.H. *An Autoregressive Distributed Lag Modelling Approach to Cointegration Analysis*; Cambridge University Press: Cambridge, UK, 1999; pp. 371–413.
21. Kondaiah, V.Y.; Saravanan, B.; Sanjeevikumar, P.; Khan, B. A review on short-term load forecasting models for micro-grid application. *J. Eng.* **2022**, *2022*, 665–689. [CrossRef]
22. Negara, H.R.P.; Syaharuddin; Kusuma, J.W.; Saddam; Apriansyah, D.; Hamidah; Tamur, M. Computing the auto regressive distributed lag (ARDL) method in forecasting COVID-19 data: A case study of NTB Province until the end of 2020. *J. Phys. Conf. Ser.* **2021**, *1882*, 012037. [CrossRef]
23. Madziwa, L.; Pillalamarry, M.; Chatterjee, S. Gold price forecasting using multivariate stochastic model. *Resour. Policy* **2022**, *76*, 102544. [CrossRef]
24. Dai, Y.; Yang, X.; Leng, M. Forecasting power load: A hybrid forecasting method with intelligent data processing and optimized artificial intelligence. *Technol. Forecast. Soc. Change* **2022**, *182*, 121858. [CrossRef]
25. Liu, S.; An, Q.; Zhao, B.; Hao, Y. A Novel Control Forecasting Method for Electricity-Carbon Dioxide Based on PID Tracking Control Theorem. In Proceedings of the 2024 4th Power System and Green Energy Conference (PSGEC), Shanghai, China, 22–24 August 2024; pp. 500–504.
26. Álvarez, V.; Mazuelas, S.; Lozano, J.A. Probabilistic Load Forecasting Based on Adaptive Online Learning. *IEEE Trans. Power Syst.* **2021**, *36*, 3668–3680. [CrossRef]
27. Åström, K.J.; Wittenmark, B. On self tuning regulators. *Automatica* **1973**, *9*, 185–199. [CrossRef]
28. Achouri, F.; Mendil, B. Wind speed forecasting techniques for maximum power point tracking control in variable speed wind turbine generator. *Int. J. Model. Simul.* **2019**, *39*, 246–255. [CrossRef]
29. Shao, K.; Zheng, J.; Wang, H.; Wang, X.; Lu, R.; Man, Z. Tracking Control of a Linear Motor Positioner Based on Barrier Function Adaptive Sliding Mode. *IEEE Trans. Ind. Inform.* **2021**, *17*, 7479–7488. [CrossRef]
30. Zhao, L.; Li, J.; Li, H.; Liu, B. Double-loop tracking control for a wheeled mobile robot with unmodeled dynamics along right angle roads. *ISA Trans.* **2023**, *136*, 525–534. [CrossRef]
31. Woodbury, M.A. *Inverting Modified Matrices*; Department of Statistics, Princeton University: Elizabeth, NJ, USA, 1950.
32. IEA. *Greenhouse Gas Emissions from Energy Highlights—Data Product*; Technical Report; IEA: Paris, France, 2024.

33. EIA. *Electricity Consumption in the United States Was About 4 Trillion Kilowatthours (kWh)*; Technical Report; EIA: Paris, France, 2022.
34. Miller, C.; Kathirgamanathan, A.; Picchetti, B.; Arjunan, P.; Park, J.Y.; Nagy, Z.; Raftery, P.; Hobson, B.W.; Shi, Z.; Meggers, F. The Building Data Genome Project 2, energy meter data from the ASHRAE Great Energy Predictor III competition. *Sci. Data* **2020**, *7*, 368. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.