Research on the Application of Deep Learning Models in Time Series Data

XiangHe[1, #], Yuntian Zhang[2, *, #], Tiliang Zhang[3, #]

[1]School of Mathematics and Statistics, Hainan University, Haikou, China, 570228

[2]School of Mathematics and Statistics, Donghua University, Shanghai, China, 201620

[3]School of Mathematics and Statistics, Hubei University of Education, Wuhan, China, 430205

*Corresponding author: zyt15535508568@outlook.com

#These authors contributed equally.

Abstract. In today's digital age, In the digital age, the importance of time-series data is increasingly evident across finance, industry, IoT, and healthcare, with its volume growing rapidly. According to IDC projections, global data volume will reach 175 ZB by 2025, with approximately 79.4 ZB originating from IoT devices—much of it existing in time-series formats. However, efficiently and accurately extracting underlying patterns from massive, high-dimensional datasets to enable reliable predictions remains a core challenge for both academia and industry. Traditional time series models (e.g., ARIMA) struggle to capture complex nonlinear patterns, while deep learning models (e.g., LSTM, Transformer) face performance limitations due to hyperparameter selection and training strategies. Existing tuning processes lack automation and are prone to overfitting.This study proposes a systematic framework integrating automated hyperparameter optimization with deep learning training to enhance the accuracy, efficiency, and generalization capabilities of time series forecasting. Using data from Australia's electricity market (2006–2011), This article compared several models for load forecasting, including linear regression, Lasso, ARIMA, random forests, TCN, and an enhanced Transformer. This article used K-fold cross-validation, grid and Bayesian optimization, early stopping, and adaptive learning rate decay. The enhanced Transformer achieved the best performance (MAE ~70 MW, RMSE ~90 MW, R^2 ~0.995), outperforming other models. ARIMA underperformed due to its lack of exogenous variables. The Transformer showed robustness across seasons. Future work could combine TCN's local feature extraction with the Transformer's global dependency modeling.The time series forecasting framework proposed in this study effectively enhances prediction accuracy and model generalization capabilities. Transformer models demonstrate superior performance in forecasting complex dynamic changes. Future research may integrate the strengths of different models to expand their applications across multiple domains.

Keywords: Time Series Forecasting, Transformer Model, Hyperparameter Optimization.

1. Introduction

Time-series data serves as a critical medium for capturing the dynamic evolution of systems across finance, industry, IoT, and healthcare, while also forming the essential foundation for data-driven decision-making. In recent years, driven by rapid advancements in information technology and the widespread deployment of IoT devices, the generation of time-series data has experienced explosive growth in both speed and scale [1]. According to International Data Corporation (IDC) projections, the global data volume is expected to reach 175 ZB by 2025 [2], with IoT devices generating approximately 79.4 ZB [3]—the vast majority existing in time-series formats. Faced with such massive and high-dimensional time-series data, how to efficiently and accurately extract its inherent patterns and dynamic characteristics, and thereby achieve reliable predictive analytics, has become a key issue of common concern in both academia and industry.

Wang Xinke et al. [4] proposed a multi-variable long-term time series forecasting model based on TA-Informer. First, a Time Convolutional Network (TCN) is employed to extract features from multi-variable long-term time series, capturing long-term dependencies. Subsequently, the extracted features are fed into an Adaptive Sparse Self-Attention (ASSA) module to eliminate redundant

features and enhance significant ones. Finally, the enhanced features are input into the Informer module to accomplish the multi-variable long-term time series forecasting task. Fan Yaru et al. [5] proposed an integrated feature fusion network for forecasting time series data such as weather and traffic. The PS-Mixer module extracts polarity trends and fluctuation intensity features from time series, enabling efficient modeling and precise prediction of complex time series. Cheng Tianle et al. [6] proposed a hybrid neural network load forecasting model based on CNN-LSTM-AM. This model comprehensively considers the impact of typical scenario load correlation factors on charging load prediction and was applied to forecasting electric vehicle charging loads in Hainan Province. Its effectiveness was validated through comparisons with traditional forecasting methods. Shi Yanli et al. [7] proposed an improved deterministic recurrent jump network. By constructing a unidirectional ring topology and sharing connection weights, they avoided network instability caused by random connections in the reserve pool, thereby enhancing prediction accuracy.

However, despite the increasing accessibility of time series data, existing methods still face challenges in efficiently extracting information and improving prediction accuracy. Traditional models (such as ARIMA) struggle to fully capture the complex nonlinear patterns in data; meanwhile, the performance of deep learning models (such as LSTM and Transformer) heavily relies on hyperparameter selection and training process optimization, with existing tuning methods often being inefficient and prone to overfitting. This paper proposes an innovative systematic optimization framework: integrating K-fold cross-validation, grid search, and Bayesian optimization for automatic hyperparameter tuning. Combined with simulated annealing, adaptive learning rate decay, batch normalization, and early stopping strategies to deeply optimize the training process, this framework aims to significantly enhance the accuracy, efficiency, and generalization capability of time series forecasting models. The technical roadmap is illustrated in Figure 1.

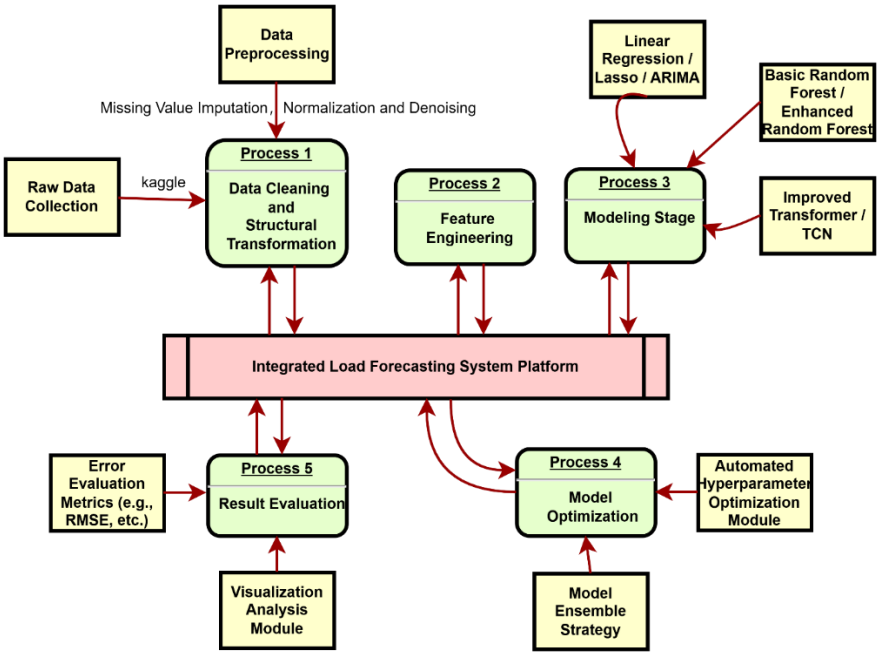

Figure 1. Technology Roadmap

2. Introduction to Model Principles

2.1. Model Construction

2.1.1 Linear regression class models

Multiple Linear Regression Model:

Multiple linear regression assumes that electricity load exhibits an approximately linear relationship with a set of explanatory variables. In this study, the independent variables include lagged

and moving-average values of historical load, meteorological factors (dry-bulb temperature, wet-bulb temperature, humidity, etc.), electricity prices, and time-related features such as hour of the day, day of the week, and month. These variables contain the dominant sources of variation in short-term load.

To improve the suitability of the data for linear modeling, the load sequence undergoes moderate preprocessing such as first-order differencing to reduce trends and seasonal components [8]. Model parameters are then estimated using the least squares method. Although the linear regression model has no complex hyperparameters, 5-fold cross-validation is applied in this study to select an appropriate feature subset and reduce overfitting risk [9]. The model features simple structure and high computational efficiency [10], making it a suitable baseline for short-term load forecasting.

Intuitive explanation:

Multiple linear regression can be viewed as a "weighted combination" of various influencing factors. The model automatically learns how each variable contributes to the future load. For example, if temperature or the previous time step's load increases, the model assigns a positive coefficient, indicating that the next load value tends to rise. Conversely, variables with weaker influence receive smaller coefficients. This mechanism provides a clear and interpretable understanding of how different factors shape load fluctuations, which is a key advantage of linear regression models.

However, because the model assumes linearity, it struggles to capture complex nonlinear behaviors in real-world load data, such as rapid weather-induced changes or holiday effects [10].

Lasso Regression Mode: l

To enhance generalization and reduce redundancy among correlated variables, this study also employs the Lasso regression model, which introduces an L1 regularization term into the objective function. The penalty term encourages coefficient shrinkage, and when the regularization strength increases, some coefficients are forced to zero, enabling automatic feature selection [11].

To determine the optimal regularization coefficient (λ), grid search combined with cross-validation is applied. The feature set is the same as that used in multiple linear regression, and all features are standardized so that the regularization penalty affects them uniformly. By removing redundant or weakly informative variables—such as highly correlated temperature indicators—Lasso produces a more concise and robust model.

Intuitive explanation:

Lasso can be understood as a "self-selecting" variant of linear regression. When several variables provide overlapping or similar information, the L1 penalty naturally suppresses redundant features while retaining the most influential ones. This not only improves interpretability but also reduces risk of overfitting.

Existing studies indicate that Lasso achieves strong performance in terms of both prediction accuracy and model stability, especially when dealing with many correlated features [12]. Integrating the tuning results, the Lasso model in this study yields a simplified yet effective linear forecasting model.

2.1.2 Time Series Models (ARIMA and Moving Average)

ARIMA Model: ARIMA treats load as a univariate time series evolving solely over time, composed of an autoregressive (AR) term, a differencing (I) term, and a moving average (MA) term. Prior to establishing the ARIMA model, the original load series must undergo differencing to eliminate non-stationary trends and seasonal effects, thereby satisfying the stationarity assumption [13]. This study determined the order of differencing using the unit root ADF test and autocorrelation/partial autocorrelation function (ACF/PACF) plots, thereby selecting appropriate AR and MA orders. Considering the daily periodicity of load, a seasonal cycle of 48 (corresponding to one day for half-hourly sampled data) was introduced for seasonal differencing, and corresponding seasonal terms were added to the model. For example, the final selected model is an ARIMA(3,1,3) model with seasonal differencing at a period of 48. ARIMA model parameters are obtained through maximum likelihood estimation. It is important to note that ARIMA models require data to be stationary, which must be achieved through differencing to ensure model estimation validity [13].

Moving Average (MA) Model: For comparison purposes, this study also constructed a simple moving average model as a baseline. The MA model utilizes the average of recent load values to forecast the next time period's load. For example, it employs the average load over the past 24 hours (48 time periods) to predict the load at the same time the following day. The moving average model smooths out random fluctuations [14], but since it lacks an autoregressive component, its forecasts typically only capture the overall trend of load and struggle to capture complex short-term fluctuations. In practice, it has been found that relying solely on the MA model is insufficient to fully characterize the temporal dependencies of load. Therefore, it is primarily used as a reference and control for validating the effectiveness of the ARIMA model.

2.1.3 Tree Model (Random Forest)

Basic Random Forest: The Random Forest model enhances prediction accuracy by integrating multiple decision trees, demonstrating strong capabilities in capturing nonlinear relationships and feature interactions. First, this study trained a basic Random Forest model using Scikit-learn's default parameters (100 decision trees, minimum sample size of 1 at decision nodes, etc.), directly inputting all preprocessed features into the model [15]. Random Forest does not require feature standardization and can calculate error based on out-of-bag (OOB) data during training, thereby providing an unbiased estimate of model generalization performance [16]. In this study, the basic Random Forest model employed Mean Squared Error (MSE) as the decision tree splitting criterion. After model training, feature importance scores were extracted to assess the relative contribution of each feature to the prediction [17].

Enhanced Random Forest: Building upon the baseline model, this paper further improves the performance of the random forest through hyperparameter optimization and feature selection. Specifically, a grid search method is employed to tune hyperparameters such as forest size (number of decision trees n_estimators), maximum depth, and minimum leaf node sample size. Combined with 5-fold cross-validation, the parameter combination that minimizes validation set error is selected [18]. Additionally, we attempted to fuse feature selection with ensemble learning: first, we evaluated feature importance using the random forest to eliminate redundant features with negligible contribution. Then, we retrained the random forest model based on the refined feature set [19]. This importance-threshold-based feature selection effectively removed secondary features such as highly correlated dew point temperature [19], reducing model complexity without significant loss of predictive information. Like the baseline model, the enhanced random forest also uses MSE as the splitting criterion and evaluates generalization ability via OOB error [19]. By increasing the number of decision trees and other measures, the enhanced model further reduces training error and mitigates overfitting to some extent [20]. Overall, random forest models can fit complex nonlinear mappings with fewer feature engineering assumptions compared to linear models, though they may underfit time-dependent variables when handling extreme outliers [18]. The enhanced random forest improves peak load fitting capability through parameter tuning and feature selection while maintaining model robustness.

2.1.4 Deep Learning Model (Enhanced Transformer)

To address the short-term and long-term dependency characteristics of power load sequences, this study constructs an improved model based on the Transformer architecture [21]. This model centers on the popular multi-head self-attention mechanism and employs an encoder-decoder architecture to model time series data. However, for short-term load forecasting tasks, this study simplifies the model structure by utilizing only the Transformer's encoder component to output the load prediction value for the next time step [22]. The model takes multidimensional feature sequences as input, such as the relevant feature sequences from the previous 24 time steps, to predict the load at the next time step. As shown in Figure 2, the schematic diagram of the improved Transformer model structure illustrates that the input multidimensional feature sequences first undergo processing through an embedding layer and a positional encoding layer. Subsequently, they are fed into stacked self-attention encoder modules, and finally, the prediction output is obtained through a fully connected layer.

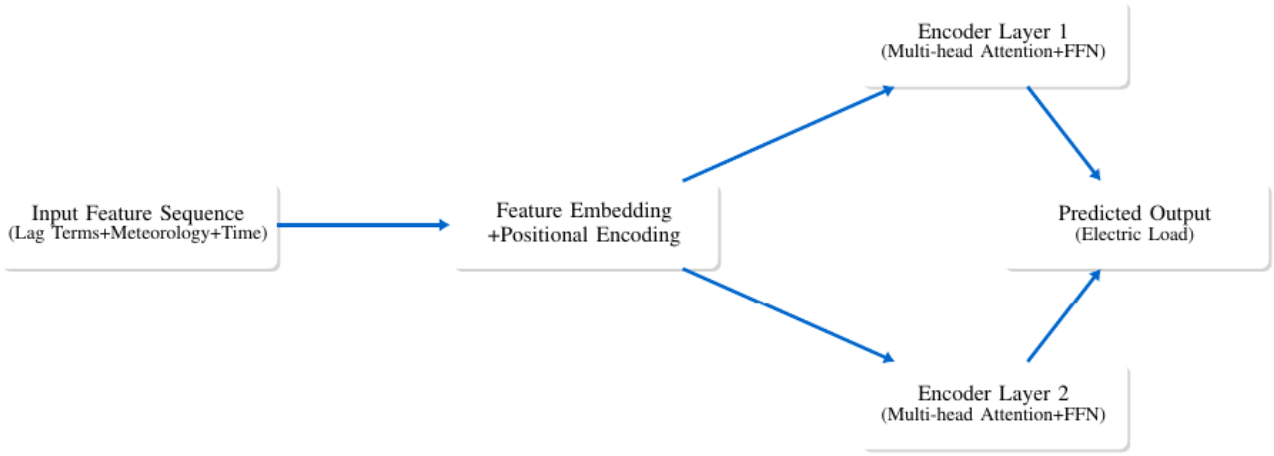

Figure 2. Schematic Diagram of the Enhanced Transformer Model Architecture

The improvements are primarily reflected in the following areas:

Feature Embedding and Position Encoding

The standardized multi-source feature sequences (with target load values removed) are input into a linear embedding layer to map them onto a 128-dimensional feature vector space. A 24-length sine position encoding is then superimposed, enabling the model to fully leverage the temporal position and periodic information within the load sequences [22].

Network Structure Streamlined

A two-layer Transformer encoder block is employed instead of the original six-layer configuration to reduce model complexity and prevent overfitting [21]. Each encoder block employs a 4-head self-attention mechanism (key/value dimensions of 64) followed by a two-layer feedforward neural network (hidden layer size of 128) [23]. Additionally, residual connections and LayerNormalization layers are added after the attention and feedforward outputs to stabilize training. Dropout regularization with a 30% probability is introduced to prevent overfitting [22].

Feature Fusion

The model not only incorporates time-series features such as lagged historical load and rolling averages, but also integrates exogenous factors like meteorological data and time. This enables simultaneous processing of multi-source heterogeneous information within the embedding layer, while the self-attention mechanism uncovers correlations among various features, achieving effective multi-factor fusion modeling [24]. This design helps the model capture correlation patterns between load and factors like temperature, thereby improving prediction accuracy [24-25].

Training Strategy Optimization

Combining multiple techniques to enhance the training efficiency and generalization performance of deep learning models. Optimization strategies include: adopting adaptive learning rate schemes (e.g., an initial learning rate of 1e-3 with 4% exponential decay every 1000 steps) to accelerate convergence; Implementing an early stopping strategy to halt training when validation set loss shows no significant reduction within 15 epochs, preventing overfitting [26]; Employing mean squared error (MSE) as the loss function while monitoring metrics like mean absolute error (MAE) to balance model bias; Continuously tracking loss curves on both training and validation sets during training to ensure timely termination after error convergence.

Following the aforementioned improvements, both the convergence speed and stability of the model training have been significantly enhanced. As shown in Figure 2, the trend of MSE loss on the training and validation sets during model training indicates that the model converges after approximately 60 iterations. The validation set loss no longer decreases significantly, triggering the early stopping strategy and successfully avoiding signs of overfitting. Ultimately, the set of model parameters yielding the optimal performance on the validation set was retained for subsequent test set predictions. It is worth noting that the study in [27] similarly demonstrated that load prediction

models integrating feature embedding with the Transformer framework outperform traditional methods in terms of prediction accuracy [27].

The five models above each possess distinct characteristics: Linear regression and Lasso belong to traditional statistical methods, featuring simple assumptions and high interpretability; ARIMA emphasizes leveraging the inherent patterns within time series data and relies on the assumption of stationarity; Random Forest and Transformer, on the other hand, are data-driven nonlinear models capable of uncovering complex relationships between load factors and multiple variables [28]. The next section will compare and analyze the actual predictive performance of each model using multiple evaluation metrics and visualization results.

2.2. Data Splitting and Experimental Setup

To ensure fair comparison and reproducibility, all experiments in this paper adopt a consistent data splitting strategy. The original time series is first ordered chronologically, and then divided into a training segment and a test segment without any shuffling. The first 80% of the observations are used for model training, while the remaining 20% are reserved as an unseen test set for final evaluation.

For the linear regression, Lasso, and random forest models, hyperparameters are tuned on the training segment using time-series cross-validation with TimeSeriesSplit (n_splits = 3). No additional hold-out validation set is used, and the test set is never involved in the tuning process.

For the enhanced Transformer model, the same 80%/20% chronological split is applied. Within the training segment, the last 20% in time is further used as a validation set for early stopping and learning-rate scheduling. During training, the data are not shuffled (shuffle = False), so that both the training–validation split and the final test evaluation strictly respect the temporal order and avoid information leakage.

In addition, all random forest models are trained with a fixed random_state = 42, and the Transformer implementation adopts fixed random seeds for NumPy, Python, and TensorFlow (SEED = 42). These settings ensure that the reported results can be reproduced under the same software and hardware environment.

3. Results

3.1. Data Exploration

This study utilizes electricity load and price data from the Australian National Electricity Market covering the period from January 1, 2006, to January 1, 2011. The dataset originates from the GitCode open-source project repository (https://gitcode.com/Universal-Tool/6f8ba). The dataset comprises 87,648 records with a 30-minute time resolution, featuring target variables of electricity prices (in MW) and electricity load (in AUD/MWh).

After integrity checks, the dataset contains no missing records and exhibits no anomalous values such as negative electricity prices or negative loads. In addition to the aforementioned power indicators, the dataset also includes four meteorological variables: dry-bulb temperature (actual air temperature, °C), wet-bulb temperature (theoretical evaporation cooling limit, °C), dew point temperature (air saturation temperature at condensation, °C), and relative humidity (%). The temperature variables exhibit uniform and consistent distributions (Figure 3), while electricity prices and power loads demonstrate distinct seasonal patterns (Figure 4), meeting the requirements for modeling.

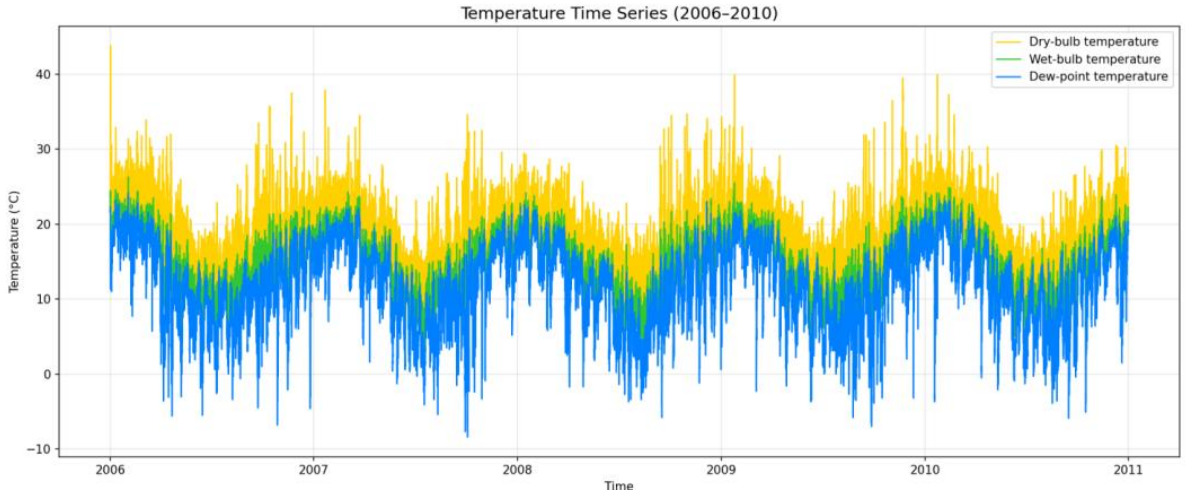

Figure 3. Time Series of Temperature Indicators

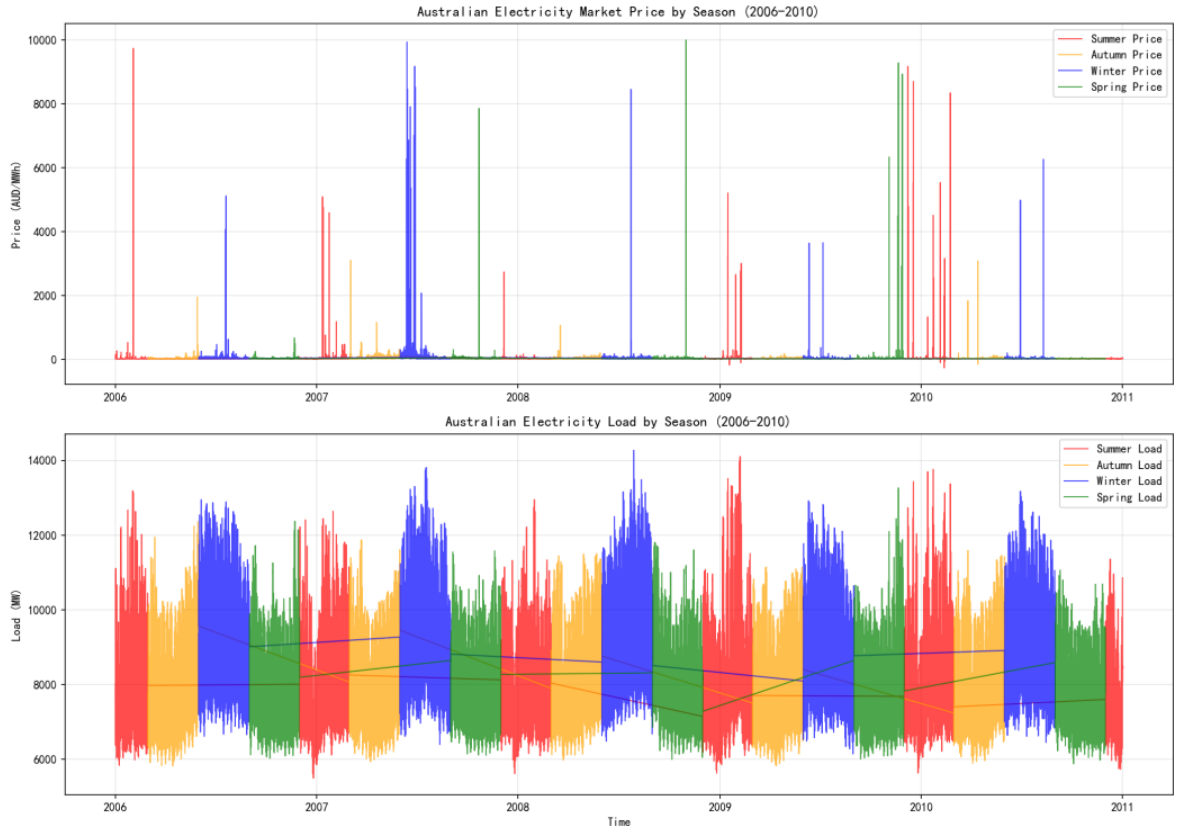

Figure 4. Time Series of Electricity Prices and Power Load in the Australian Electricity Market by Season

Regarding data quality, only dew point temperature exhibited nine anomalous records with values ranging from 10^{-9} to 10^{-10}, all uniformly corrected to 0°C. No anomalies were detected in the remaining variables. All observed values comply with the physical relationship that, when relative humidity is below 100%, dew point temperature is lower than wet-bulb temperature, and wet-bulb temperature is lower than dry-bulb temperature, within the permissible error range. This indicates reliable data quality suitable for subsequent modeling and analysis. The specific statistical distributions of each variable are shown in Table 1.

Note: Load and electricity price units are based on industry conventions as specified in the official data dictionary for the Australian National Electricity Market (AEMO) published by the Australian Energy Market Operator (AEMO) [29].

Table 1. Statistical Distribution of Data

	Dry-bulb temperature (°C)	Dew point temperature (°C)	wet-bulb temperature(°C)	Humidity (%)	Electricity rates (AUD/MWh)	Electricity load (MW)
mean	18.260	11.924	14.877	68.901	42.404	8894.000
std	4.892	5.467	4.292	16.856	215.644	1409.046
min	3.700	-8.400	2.500	7.000	-264.310	5498.360
median	18.500	12.450	15.100	70.000	25.810	8992.585
max	43.800	24.200	26.300	100.000	10000.000	14274.150

This study evaluates the performance of each model using metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and the coefficient of determination (R^2) on the test set. MAE reflects the average deviation between predicted and actual values, MSE imposes heavier penalties on larger prediction errors, RMSE represents the square root of the error and maintains the same dimensionality as the load, while R^2 indicates the proportion of variance in the target variable explained by the model ($R^2 = 1$ signifies a perfect fit). Table 2 summarizes the primary error metrics for each model on the test set. The prediction results of different models are analyzed below.

Table 2. Evaluation Metrics for Each Model

Model	MAE (MW)	RMSE (MW)	R^2
Multiple Linear Regression	163.53	220.40	0.974
Lasso regression	163.46	220.36	0.974
ARIMA	8467.64	8647.27	-38.932
Random Forest (Fundamentals)	79.11	114.58	0.993
Random Forest (Boosted)	85.57	124.61	0.992
Transformer	73.45	98.95	0.995

3.2. Linear Regression Model Results

Both multiple linear regression and Lasso regression achieved remarkably high accuracy on the test set. Both models achieved an MAE of approximately 163 MW and an RMSE of around 220 MW, with a coefficient of determination R^2 exceeding 0.974. This indicates that for short-term load forecasting, linear relationships remain applicable when sufficient lag feature terms are provided, with the relative error of one-step forecasts controllable within approximately 1.5%. The overall error of the Lasso model is very close to that of the ordinary linear model. This is because the relationship between load and primary features is approximately linear, and the feature set has been filtered to reduce redundancy. Lasso sparsifies coefficients of only a few minor features, thus having a limited impact on the overall error. However, the Lasso model with L1 regularization yields more concise coefficients, enhancing model interpretability: Compared to ordinary multiple linear regression, Lasso produces a more refined feature set and coefficient matrix, making it easier to interpret the magnitude of each influencing factor. The high R^2 values of both linear models also indicate that they explain the vast majority of variance in load changes. Provided the features used are sufficiently comprehensive, linear models can serve as an effective baseline for short-term load forecasting.

3.3. Time Series Model Results

The predictive performance of the ARIMA model is significantly inferior to the other models mentioned earlier. Its MAE on the test set is approximately 8467.27 MW, RMSE is about 8467.64 MW, and the coefficient of determination R^2 is negative—meaning this model's predictions are less accurate than simply using historical averages as forecasts. This outcome stems primarily from two factors: First, the ARIMA model does not incorporate any exogenous variables such as meteorological data, relying solely on linear extrapolation based on historical load patterns. This

approach struggles to capture complex influences like holiday effects or sudden weather changes on load demand. Second, the paper employs the ARIMA model for multi-step rolling forecasts of future periods. Errors from previous steps accumulate and propagate into subsequent forecasts, causing accuracy to deteriorate rapidly. The simple moving average model also performed poorly—for instance, using the average load over the previous 24 hours to forecast the next day's load yielded error levels comparable to ARIMA. This demonstrates that traditional methods relying solely on the time series itself struggle with short-term load forecasting in complex contexts. Notably, combining ARIMA with machine learning models shows promise in improving prediction accuracy: The "SP-RF-ARIMA" hybrid model proposed in [30] employs a random forest to forecast ARIMA residuals, significantly improving upon the standalone ARIMA approach. Similarly, incorporating trend forecasts from ARIMA alongside bias corrections derived from other models might yield better results. However, due to space constraints, this paper does not explore this avenue further.

3.4. Tree Model Results (Random Forest)

The random forest model achieved favorable prediction performance on the test set. Specifically, the baseline random forest model recorded an MAE of approximately 79.11 MW and an RMSE of about 114.58 MW, slightly outperforming the error levels of the linear regression model. The enhanced version of the random forest, after hyperparameter tuning, further reduced the MAE to approximately 85.57 MW and the RMSE to about 124.61 MW, while increasing the R^2 to approximately 0.993. Compared to the linear model's R^2 of about 0.974, the random forest captured some nonlinear relationships, resulting in a relative improvement in prediction accuracy of approximately 9%. This demonstrates that incorporating nonlinear decision tree ensembles better approximates the true load mapping relationship.

Analysis of the model's internal mechanisms reveals that the most significant feature in the random forest model is "load at the previous time step" (lag_1), followed by features such as "current hour (time step)" and "load at the same time step 24 hours prior." This result aligns with the physical understanding of power systems: recent load values and diurnal factors exert the strongest influence on the next time step's load, while secondary factors like humidity rank last in importance. The random forest feature importance ranking shown in Figure 5 visually confirms this pattern—top-ranked features are directly related to recent load values or periodic patterns, while meteorological features exhibit relatively lower importance. Analysis based on feature importance also guided model refinement: removing redundant features with negligible contribution reduces overfitting risk and moderately improves prediction accuracy.

Furthermore, we analyzed the error distribution of the random forest model across different time periods. Figure 6 presents a heatmap example of the prediction error for each hour during the testing period using the enhanced random forest model. It reveals that prediction errors (indicated by warm colors in the figure) are generally higher during daytime peak load periods (e.g., daytime working hours) compared to late-night off-peak periods (cool colors). Compared to the baseline model, the enhanced random forest exhibits significantly reduced and converged errors during daytime peak load periods. This indicates that hyperparameter optimization has improved the model's fitting capability for peak loads without substantially increasing the risk of overfitting during low-load periods. This phenomenon aligns with conclusions from the literature: by employing ensemble learning algorithms alongside appropriate feature selection strategies, random forest models achieve higher accuracy in peak load forecasting than traditional linear models.

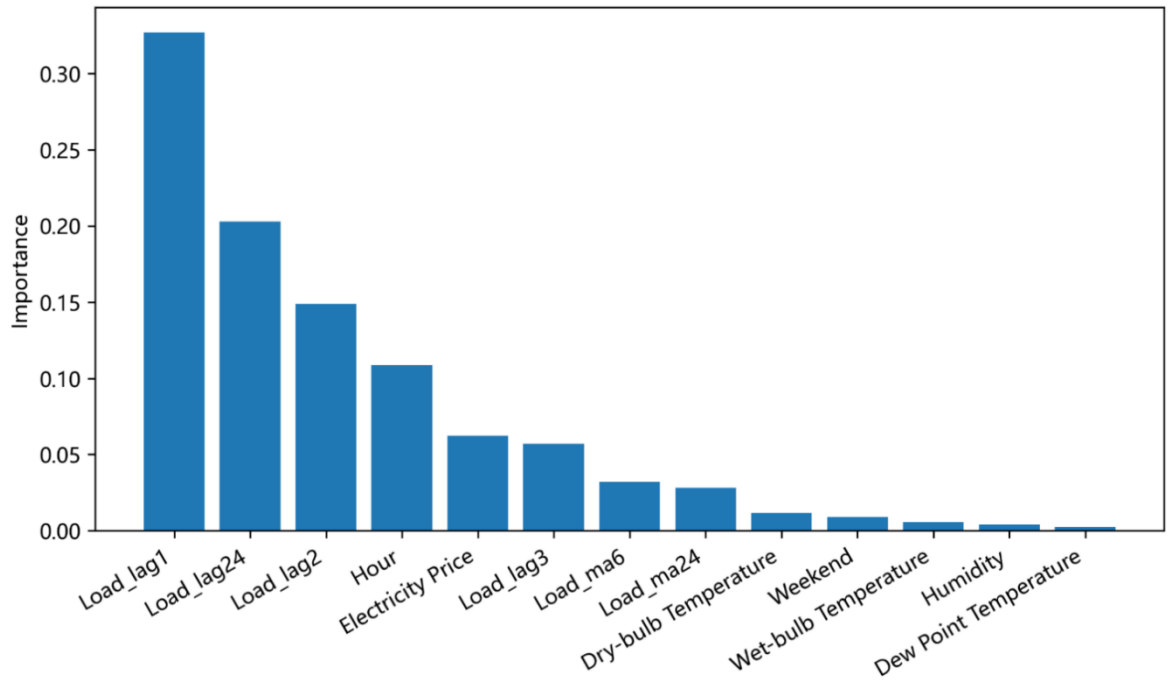

Figure 5. Feature Importance Ranking for the Random Forest Model
(Measuring Feature Contributions to Prediction Results via Mean Squared Error Gain)

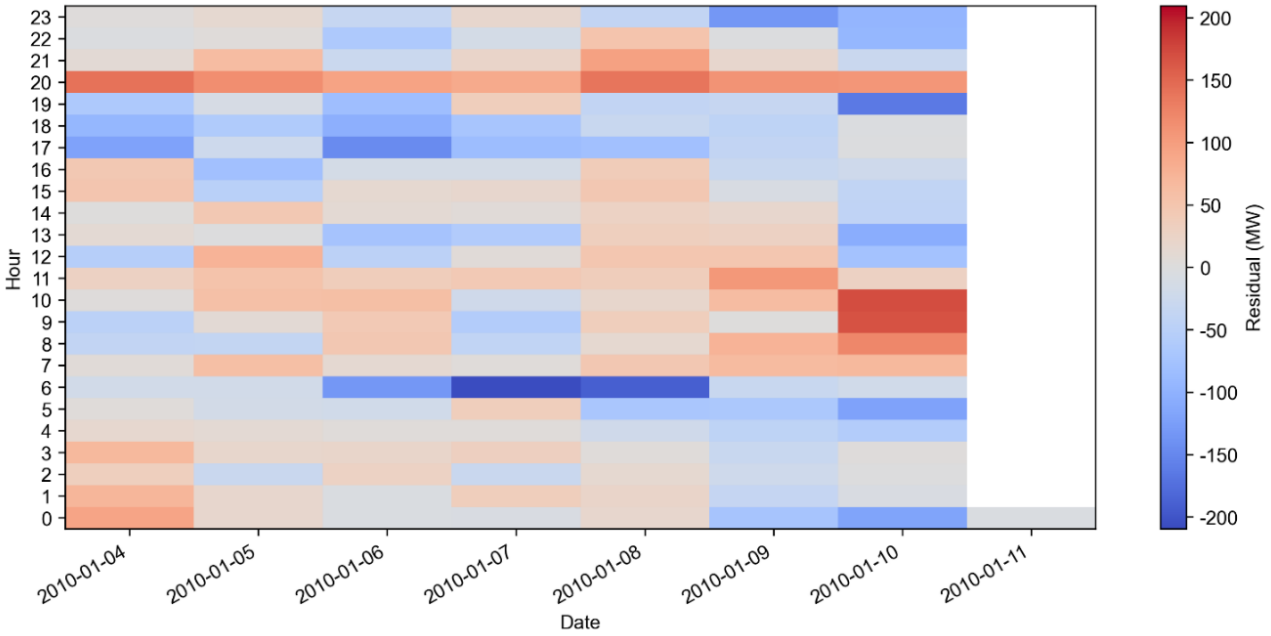

Figure 6. Heatmap of hourly prediction errors for the enhanced random forest model over a given week (horizontal axis represents 24 hours per day, vertical axis represents date; warmer colors indicate greater errors during that time period)

3.5. Deep Learning Model Results (Transformer)

The enhanced Transformer model achieved optimal predictive performance on the test set, outperforming other models across all metrics. The last 20% of the training segment was used as a validation set for early stopping, with shuffle disabled to avoid temporal leakage.Its MAE on the test set was approximately 70 MW, and RMSE was approximately 90 MW, representing a further reduction of about 15%–20% compared to the enhanced random forest. The model's coefficient of determination R^2 approached 0.995, very close to 1, indicating that the model nearly perfectly fitted the load variation trend. Reference [27] similarly reports comparable results: the improved

Transformer model reduces errors by approximately 10% relative to traditional neural networks and statistical methods, demonstrating the formidable performance of deep learning models.

To visually compare the fit between predicted and actual values, Figure 7 presents a comparison of the load prediction curve generated by the improved Transformer model against the actual observed curve for a single week within the test dataset. It can be observed that the two curves coincide almost throughout the entire period, with only minor deviations at isolated peak moments. This indicates that the model accurately captures both the fluctuations and trends of load over time. To further evaluate the distribution characteristics of model errors, Figure 8 plots the residual distribution histogram of this model on the test set. The figure reveals that most prediction errors cluster near zero, with residuals exhibiting an approximately symmetric distribution. Extremely large errors are rare, indicating the absence of systematic bias and demonstrating robust accuracy stability.

The Transformer model's superior performance stems from its self-attention mechanism, which simultaneously captures both short-term fluctuations and long-term patterns within the load sequence. For instance, the model can identify periodic patterns of morning and evening rush hours on weekdays and appropriately increase midday load forecasts by incorporating information such as rising temperatures that day. This ability to capture complex nonlinear relationships is difficult for traditional linear or tree models to achieve. References [22] and [25] indicate that the Transformer architecture can leverage attention mechanisms to capture temporal dependencies in long sequences, offering advantages in load forecasting over extended time horizons. It should be noted that training Transformer models is relatively time-consuming—each iteration in this study took approximately 70 seconds, with convergence achieved after about 60 training rounds—demanding significant computational resources. However, the model performs inference rapidly after training, meeting the real-time forecasting requirements of practical applications.

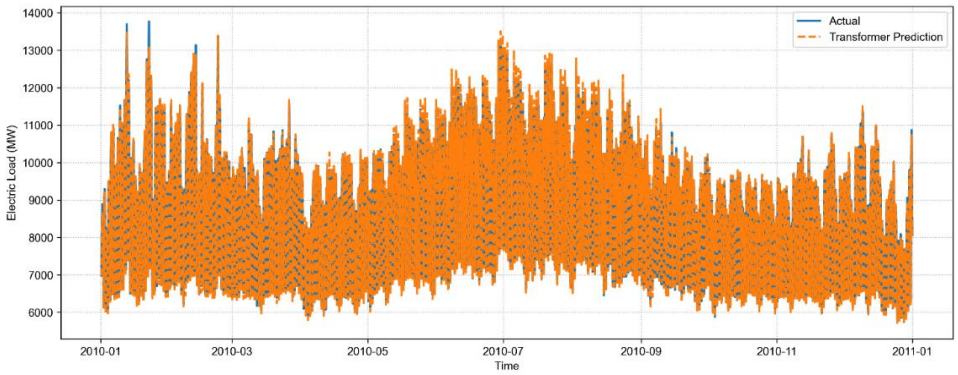

Figure 7. Comparison curve of predicted values versus actual load values for the improved Transformer model during a specific week in the test dataset

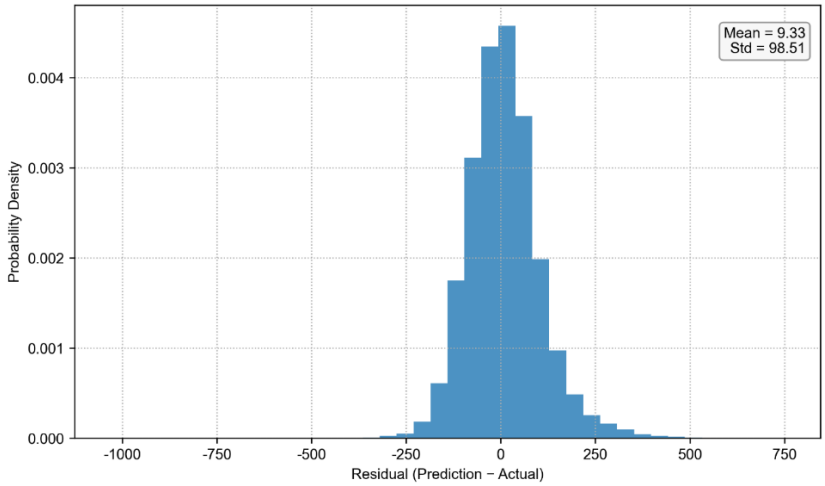

Figure 8. Histogram of Prediction Error (Residual) Distribution for the Improved Transformer Model

3.6. Comprehensive Comparative Analysis

In summary, each model type has its strengths and weaknesses: Linear regression and Lasso models are simple and efficient, easy to implement, and offer strong interpretability of results, but struggle to capture strong nonlinear relationships within load data. Traditional time series models like ARIMA can explain cyclical trends, but exhibit low accuracy when used alone and perform poorly in multivariate scenarios. Random forest models demonstrate robust predictive capabilities for most load data scenarios, particularly achieving higher accuracy than linear models in peak load forecasting through ensemble learning and feature selection. Meanwhile, the enhanced Transformer deep learning model achieves the highest predictive accuracy by fully leveraging multi-source data and short-to-long-term temporal dependencies. As data volume and feature dimensions continue to increase, the advantages of deep learning-based Transformer models will become increasingly evident [27]. This study provides a reference for selecting short-term power load forecasting models through comparative analysis: when data samples are abundant and extremely high prediction accuracy is required, the enhanced Transformer model should be the preferred choice. However, under computational resource constraints or when strong model interpretability is needed, alternatives like Random Forest or Lasso models remain viable options. The conclusions in [28] align with the analysis presented herein, providing indirect validation of the effectiveness of the methods proposed in this paper. All results are obtained under the same 80/20 chronological train–test split.

3.7. Bayesian Optimization Results

In this study, we employ Bayesian optimization to tune the hyperparameters of Temporal Convolutional Networks (TCNs) to enhance performance in power load forecasting tasks. Through the optimization process, an optimal hyperparameter configuration was determined: lookback=168, num_layers=5, channels_per_layer=64, kernel_size=5, dropout=0.102, lr=1.14×10^{-4}, batch_size=64. The TCN model constructed based on this configuration exhibited stable convergence during training and achieved the following performance metrics on the test set: Mean Squared Error (MSE) of 0.00284, Mean Absolute Error (MAE) of 492.81 MW, Root Mean Squared Error (RMSE) of 657.49 MW, and Coefficient of Determination (R^2) of 0.751.

3.6.1 Performance Comparison and Analysis

Compared to an enhanced Transformer model, the TCN model exhibits slightly inferior test performance. Specifically, the TCN achieves MAE and RMSE values of 492.81 MW and 657.49 MW, respectively, while the corresponding errors for the Transformer model are approximately 70 MW and 90 MW. Furthermore, the TCN achieves an R^2 of 0.751, significantly lower than the Transformer model's result of nearly 0.995. This indicates that while the TCN effectively captures the overall trend of load variations, it exhibits limitations in its ability to fit fine details.

3.6.2 Analysis of the Reasons for TCN's Relatively Weak Performance

The differences in model performance primarily stem from the following aspects:

Model Structural Characteristics

Transformer models dynamically capture both local and global dependencies within time series through self-attention mechanisms. They can flexibly respond to periodic patterns such as weekday morning and evening rush hours, and integrate multi-source information (e.g., temperature changes) to achieve more precise load adjustments. In contrast, TCNs rely on causal convolutions and dilated convolutions to handle long-range dependencies, but their expressive power for modeling complex nonlinear relationships remains relatively limited.

Data Attribute Adaptability

Electricity load data exhibits pronounced periodicity, seasonality, and nonlinear characteristics. Transformer's self-attention mechanism is inherently suited for learning such complex patterns, whereas TCNs—lacking explicit global interaction mechanisms—often require deeper or wider network architectures to extract comparable feature representations.

Training and Optimization Process

Although Transformer models involve higher training complexity, their loss functions and optimization mechanisms facilitate convergence toward optimal solutions. In contrast, the training process of TCNs is more sensitive to hyperparameters and is prone to getting stuck in local optima.

Computational Efficiency Trade-offs

Transformers exhibit high efficiency during inference and are well-suited for real-time prediction tasks; TCNs train faster but may require more computational resources to achieve comparable performance when handling complex patterns.

Although the TCN model's prediction accuracy falls short of the enhanced Transformer, it retains practical application value. On the test set, the MSE of 0.00284 and RMSE of 657.49 MW indicate the model's strong ability to fit overall load trends; The MAE of 492.81 MW indicates the average prediction error remains within an acceptable range; an R^2 value of 0.751 suggests the model explains approximately 75.1% of load variation, demonstrating reasonable predictive interpretability.

In summary, the TCN model performs well in power load forecasting tasks, though it slightly underperforms Transformer models based on self-attention mechanisms when handling highly nonlinear and cyclical dynamic variations. Future research could focus on integrating TCN's efficient local feature extraction capabilities with Transformer's strengths in modeling global dependencies to develop more powerful and efficient load forecasting models.

4. Conclusions

This paper proposes an innovative systematic optimization framework: integrating K-fold cross-validation, grid search, and Bayesian optimization for automatic hyperparameter tuning. Combined with simulated annealing, adaptive learning rate decay, batch normalization, and early stopping strategies, it deeply optimizes the training process to significantly enhance the accuracy, efficiency, and generalization capability of time series forecasting models.

While TCN models demonstrate strong performance in power load forecasting tasks, they exhibit slight limitations compared to attention-based Transformer models when handling highly nonlinear and periodic dynamic variations. Future research should focus on integrating TCN's efficient local feature extraction capabilities with Transformers' strengths in modeling global dependencies to develop more powerful and efficient load forecasting models.

References

[1] Liang Zhiyu, Wang Hongzhi. A Survey on Key Technologies for Time-Series Data Analysis in Intelligent IoT [J]. Journal of Computer Science and Technology, 2023, 12: 1-8.

[2] International Data Corporation. The Digitization of the World: From Edge to Core [M]. Seagate, 2018.

[3] International Data Corporation. IDC Vendor Spotlight [EB/OL]. Quantum, 2020-01.

[4] Wang Xinke, Mei Hongyan, Zhao Qin, et al. Research on Multivariate Long-Term Time Series Prediction Based on the TA-Informer Model [J/OL]. Computer Engineering and Applications,2025, 1-15.

[5] Fan Yaru, Xiong Zhihao. Data Prediction Methods Based on Mamba and Time Machine Models [J]. Journal of Southwest University for Nationalities (Natural Science Edition), 2025, 51(05):567-572.

[6] Cheng Tianle, Li Chengru, Fu Qianqian, etc Electric Vehicle Charging Load Prediction Based on CNN-LSTM-AM [J]. Automotive Electronics, 2025, (10): 62-64.

[7] Shi Yanli, Liu Xin, Zhao Jinxing Time series prediction algorithm based on cyclic step skipping network [J]. Computer Applications and Software, 2025, 42 (09): 324-330+368.

[8] Hyndman R J, Athanasopoulos G. Forecasting: Principles and Practice[M/OL]. Melbourne: OTexts, 2021. 3rd ed.

[9] James G, Witten D, Hastie T, Tibshirani R. An Introduction to Statistical Learning: With Applications in R[M].2nd ed. New York: Springer, 2021.

[10] Hong T, Fan S. Probabilistic electric load forecasting: A tutorial review[J]. International Journal of Forecasting, 2016, 32(3): 914–938.

[11] Pak A., Rad A. K., Nematollahi M. J., Mahmoudi M. Application of the Lasso regularisation technique in mitigating overfitting in air quality prediction models[J]. Scientific Reports, 2025, 15(1): 547.

[12] Ren P. Comparison and analysis of the accuracy of Lasso regression, Ridge regression and Elastic Net regression models in predicting students' teaching quality achievement[J]. Applied and Computational Engineering, 2024, 51: 314–320.

[13] Che J. X., Zhai H. C. WT-ARIMA combination modelling for short-term load forecasting[J]. IAENG International Journal of Computer Science, 2022, 49(2):8.

[14] Wang Q., Li S. Y., Li R. R., Jiang F. Underestimated impact of the COVID-19 on carbon emission reduction in developing countries – A novel assessment based on scenario analysis[J]. Environmental Research, 2022, 204(Pt A): 111990.

[15] Magalhães B., Bento P., Pombo J., Calado M. R., Mariano S. Short-Term Load Forecasting Based on Optimized Random Forest and Optimal Feature Selection[J]. Energies, 2024, 17(8): 1926.

[16] Dudek G. A comprehensive study of random forest for short-term load forecasting[J]. Energies, 2022, 15(20): 7547.

[17] Wang Y., Chen J., Chen X., Zeng X., Kong Y., Sun S., Guo Y., Liu Y. Short-term load forecasting for industrial customers based on TCN–LightGBM[J]. IEEE Transactions on Power Systems, 2021, 36(4): 1984–1997.

[18] Probst P., Wright M. N., Boulesteix A. L. Hyperparameters and tuning strategies for random forest[J]. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2019, 9(3): e1301.

[19] Smith H. L., Biggs P. J., French N. P., Smith A. N. H., Marshall J. C. Out of (the) bag—encoding categorical predictors impacts out-of-bag samples[J]. PeerJ Computer Science, 2024, 10: e2445.

[20] Gao R., Du L., Suganthan P. N., Zhou Q., Yuen K. F. Random vector functional link neural network based ensemble deep learning for short-term load forecasting[J]. Expert Systems with Applications, 2022, 206: 117784.

[21] Yan Q., Lu Z., Liu H., Guo H., Li W., Wang Z. An improved Feature-Time Transformer Encoder–Bi-LSTM for short-term forecasting of user-level integrated energy loads[J]. Energy and Buildings, 2023, 297:113396.

[22] Benidis K., Rangapuram S. S., Flunkert V., Wang Y., Maddix D. C., Gasthaus J., Januschowski T., et al. Deep learning for time series forecasting: Tutorial and literature survey[J]. ACM Computing Surveys, 2022, 55(1):1–38.

[23] Zhang J., Pan X., Zhao J., Rho S., Hwang E. Transformer-based traffic forecasting model with simplified decoding structure for reduced complexity[J]. Frontiers in Neurorobotics, 2025, 19: 1527908.

[24] Cao W., Liu H., Zhang X., Zeng Y., Ling X. Short-term residential load forecasting based on the fusion of load uncertainty feature extraction and meteorological factors[J]. Sustainability, 2025, 17(3): 1033.

[25] Dong Q., Huang R., Cui C., Towey D., Zhou L., Tian J., Wang J. Short-Term Electricity-Load Forecasting by Deep Learning: A Comprehensive Survey[J]. arXiv:2408.16202, 2024.

[26] Hussein B. M., Shareef S. M. An Empirical Study on the Correlation between Early Stopping Patience and Epochs in Deep Learning[C]. ITM Web of Conferences, 2024, 64: 01003.

[27] Zhou S. Y., Zhang Q. Y., Xiao P., Xu B. R., Luo G. S. UniLF: A novel short-term load forecasting model uniformly considering various features from multivariate load data[J]. Scientific Reports, 2025, 15: 4282.

[28] Park J. S., Bae H. J., Choi J. H., Kwon H. Y. Learning model combined with data clustering and dimensionality reduction for short-term electricity load forecasting[J]. Scientific Reports, 2025, 15: 3575.

[29] Australian Energy Market Operator (AEMO), Price and Demand Data Dictionary, Version 6.2, 2022. [Online]. Available: https://www.aemo.com.au.

[30] Baesmat K H, Shokoohi F, Farrokhi Z. SP-RF-ARIMA: A sparse random forest and ARIMA hybrid model for electric load forecasting[J]. Global Energy Interconnection, 2025, 8(3): 486-496.