scientific reports

Check for updates

OPEN

A comparative evaluation of gradient-based optimization algorithms for short-term load forecasting using deep residual networks

Junchen Liu[1], Faisul Arif Ahmad[1]✉, Khairulmizam Samsudin[1], Fazirulhisyam Hashim[1] & Mohd Zainal Abidin Ab Kadir[2]

Short-Term Load Forecasting (STLF) is essential for the reliable and economic operation of modern power systems. Deep Residual Networks (DRNs) have emerged as an effective framework for STLF due to their ability to model nonlinear and multi-scale load patterns. Although numerous DRN-based extensions have been proposed through architectural refinement and feature enhancement, the role of gradient-based optimization algorithms in DRN-based STLF has received limited systematic investigation. Most existing studies rely on the Adaptive Moment Estimation (Adam) algorithm as the default optimization strategy, without comprehensively examining alternative gradient-based optimizers. To address this gap, this study conducts a hypothesis-driven comparative evaluation of representative gradient-based optimization algorithms within a unified DRN-based STLF framework across both temperate (ISO-NE) and tropical (MyPJ) climatic conditions. Both the original DRN, which primarily incorporates temperature as the meteorological input, and its enhanced variant, the Principal Component Analysis–Deep Residual Network (PCA-DRN), which integrates multiple weather variables through PCA, are investigated using real-world electricity load datasets. Forecasting performance is evaluated using multiple metrics, with Mean Absolute Percentage Error (MAPE) as the primary criterion, and statistical significance is assessed through a nonparametric bootstrap resampling procedure. The results demonstrate that optimizer selection significantly influences training stability and forecasting accuracy. AMSGrad achieves the most consistent performance within the original DRN across climatic conditions, whereas under PCA-based feature representation the relative advantage shifts, indicating that meteorological feature compression reshapes the optimization landscape. Overall, the findings highlight the importance of systematic optimizer evaluation and feature-representation strategies for enhancing the reliability and stability of DRN-based STLF.

Keywords Short-term load forecasting, Deep residual networks, Gradient-based optimization algorithms, Principal component analysis, Meteorological feature representation

Abbreviations

Adadelta	Adaptive delta
Adagrad	Adaptive gradient
Adam	Adaptive moment estimation
Adamax	Adam variant based on the infinity norm
AdamW	Adam with decoupled weight decay
AdaBelief	Adaptive belief optimizer
AI	Artificial intelligence
AMSGrad	Adam variant employing a maximum second-moment estimator

[1]Department of Computer and Communication Systems Engineering, Faculty of Engineering, Universiti Putra Malaysia (UPM), Serdang 43400, Selangor, Malaysia. [2]Advanced Lightning, Power and Energy Research Centre (ALPER), Faculty of Engineering, Universiti Putra Malaysia (UPM), Serdang 43400, Selangor, Malaysia. ✉email: faisul@upm.edu.my

ANN	Artificial neural network
BiGRU	Bidirectional gated recurrent unit
BiLSTM	Bidirectional long short-term memory
CI	Confidence interval
CNN	Convolutional neural network
CNN–LSTM–MMA	CNN–LSTM with multi-modal attention
Conv1D	One-dimensional convolution
DL	Deep learning
DNN	Deep neural network
DRN	Deep residual network
ERN	Ensemble residual network
FC	Fully connected
GRU	Gated recurrent unit
ISO-NE	New England independent system operator
LF	Load forecasting
LSTM	Long short-term memory
LTLF	Long-term load forecasting
MAE	Mean absolute error
MAPE	Mean absolute percentage error
MSE	Mean square error
MTLF	Medium-term load forecasting
MW	Megawatt
MyPJ	Malaysia Petaling Jaya
Nadam	Nesterov-accelerated adaptive moment estimation
NAG	Nesterov accelerated gradient
NMSE	Normalized mean square error
PCA-DRN	Principal component analysis–deep residual network
p-value	Probability value
R	Correlation coefficient
R^2	Coefficient of determination
ReLU	Rectified linear unit
ResBlock	Residual block
ResNet	Residual network
RAdam	Rectified adam
RNN	Recurrent neural network
RMSProp	Root mean square propagation
SD	Standard deviation
SELU	Scaled exponential linear unit
SGD	Stochastic gradient descent
SGD-Momentum	Stochastic gradient descent with momentum
STLF	Short-term load forecasting
Tanh	Hyperbolic tangent

In order to ensure the reliability and operational efficiency of modern power systems, Short-Term Load Forecasting (STLF) plays a critical role. Accurate prediction of electrical demand within horizons ranging from 24 h to one week supports generation scheduling, unit commitment, and energy trading decisions, ultimately improving grid stability and reducing operational costs[1]. Even a modest improvement in forecasting accuracy can lead to substantial economic benefits for utilities, highlighting the strategic importance of developing reliable STLF frameworks[2,3].

Existing STLF approaches can generally be categorized into traditional statistical methods and artificial intelligence (AI)-based models. Conventional statistical and machine learning techniques provided early insights into demand modeling; however, their ability to capture nonlinear and highly dynamic load patterns remains limited, particularly when dealing with high-dimensional inputs and complex temporal variations[4–6]. These limitations motivated the adoption of AI-driven models, especially artificial neural networks (ANNs), which improved nonlinear representation capability but were still constrained by shallow structures and susceptibility to overfitting when network complexity increased[7–10].

With the rapid development of deep learning (DL), Deep Neural Networks (DNNs) have gradually replaced shallow ANN architectures in STLF research. By enabling hierarchical feature extraction and learning complex temporal relationships, DNN-based models significantly enhanced forecasting capability[11]. Representative architectures include Convolutional Neural Networks (CNNs), which capture localized temporal patterns but struggle with long-range dependencies[12,13]; Recurrent Neural Networks (RNNs) such as Long Short-Term Memory (LSTM), which model sequential dynamics but suffer from limited computational efficiency[14,15]; and Transformer-based models, which leverage self-attention for long-term dependency modeling at the expense of higher computational cost and training instability[16–18]. Hybrid frameworks combining convolutional, recurrent, and attention mechanisms have also been explored to improve generalization performance[19,20]. Despite these advancements, challenges related to training stability, scalability, and optimization efficiency remain central issues in DL-based STLF.

To address these challenges, residual learning has emerged as a promising solution. Residual architectures introduce identity shortcut connections that facilitate gradient propagation, enabling deeper networks to be

trained more effectively. Inspired by the residual network (ResNet) architecture[21], Chen et al.[22] proposed the Deep Residual Network (DRN) for STLF, which demonstrated improved convergence stability and forecasting accuracy compared with earlier DL models. Since then, DRN-based frameworks have gained increasing attention as a reliable backbone for high-performance STLF, shifting research focus from shallow or hybrid architectures toward deeper residual learning strategies. Following the original DRN formulation, numerous extensions have been proposed to enhance model robustness and predictive performance, including task-specific DRN structures[23], ensemble-based residual frameworks[24,25], adaptive feature-driven DRN designs[26], multi-scale and inception-inspired residual modules[27,28], and hybrid DRN architectures integrating recurrent or attention mechanisms[29–31], as well as the Principal Component Analysis–Deep Residual Network (PCA-DRN) that incorporates multidimensional meteorological information through dimensionality reduction[32]. However, despite extensive architectural innovation, most DRN-based STLF studies adopt the Adaptive Moment Estimation (Adam) optimizer as a default training strategy, and the influence of alternative gradient-based optimization algorithms on DRN performance has received limited systematic investigation. This imbalance indicates that, while structural improvements in DRN have been widely explored, the role of training strategies—particularly optimizer selection—remains an under-examined yet potentially critical factor for advancing DRN-based STLF.

Despite these architectural and feature-level advancements, most existing DRN-based STLF studies continue to adopt the Adam optimization algorithm as a default training strategy, while the potential impact of alternative gradient-based optimization algorithms has rarely been systematically examined. Given the depth and complexity of DRN architectures, together with the non-stationary and highly nonlinear characteristics of electricity load time series, optimizer selection may play a critical role in determining training stability, convergence behavior, and final forecasting accuracy. However, this issue remains largely underexplored in the context of DRN-based STLF.

Furthermore, the recent development of PCA-based DRN variants introduces additional complexity to the optimization process. Unlike the original DRN, which primarily incorporates temperature as the sole meteorological input, PCA-DRN integrates multiple meteorological variables through PCA. By compressing multidimensional weather information into a compact feature space, PCA fundamentally alters the statistical properties of model inputs and, consequently, the gradient dynamics during training. Whether an optimizer that performs well for the original DRN remains optimal when PCA-transformed meteorological features are employed has not yet been investigated. This gap highlights the necessity of a systematic and controlled evaluation of gradient-based optimization algorithms under different feature-representation settings within DRN-based STLF frameworks. To explicitly examine these research gaps, this study is guided by two hypotheses: (a) optimizer choice significantly impacts the training stability and forecasting performance of DRN-based STLF models; and (b) this impact is modulated by the feature-representation strategy, particularly PCA-based meteorological compression, resulting in a non-uniform optimization performance landscape across different climatic conditions.

The main contributions of this study are summarized as follows:

1. A systematic comparative evaluation of representative gradient-based optimization algorithms is conducted for DRN-based STLF, revealing the significant influence of optimizer selection on training behavior and forecasting performance.

2. The interaction between optimizer effectiveness and feature representation is investigated by jointly analyzing the original DRN and its PCA-enhanced variant, demonstrating that PCA-based meteorological feature compression reshapes the optimization landscape.

3. The reliability of the observed performance differences is rigorously validated using a nonparametric bootstrap resampling procedure, confirming that the improvements across optimizers and model variants are statistically significant rather than attributable to random variation.

The remainder of this paper is organized as follows: Sect. 2 reviews the research background, including DL methods for STLF, DRN-based forecasting models, and gradient-based optimization algorithms. Section 3 describes the research methodology, covering data preprocessing, model architectures, experimental design, and evaluation metrics. Section 4 presents and discusses the experimental results, including benchmarking comparisons, optimizer performance analysis, and statistical significance testing. Finally, Sect. 5 concludes the paper and outlines potential directions for future research.

Research background

Deep learning methods for short-term load forecasting

The rapid advancement of sensor technologies, smart metering systems, and high-performance computing platforms has accelerated the adoption of DL techniques in STLF. Owing to their strong nonlinear representation capability and ability to capture complex temporal relationships from large-scale data, DL models have gradually replaced traditional statistical and shallow learning approaches in many forecasting applications. Existing DL-based STLF studies mainly explore convolution-oriented models, recurrent neural networks, attention-based architectures, and hybrid frameworks, each emphasizing different characteristics of load time-series patterns.

CNNs have been widely used to extract local temporal features and short-range patterns in load data. For example, Li et al.[12] reformulated load sequences into image-like representations to exploit spatial correlations through convolutional kernels, while Jurado et al.[13] introduced an encoder–decoder CNN framework enhanced with Monte Carlo Dropout to improve uncertainty estimation. Recurrent neural networks, particularly LSTM models, remain one of the most commonly adopted architectures for sequential load modeling. Narayan and Hipel[14] demonstrated that deep LSTM-based forecasting models could outperform conventional statistical approaches in regional load prediction, and Bento et al.[15] further improved forecasting accuracy by optimizing LSTM hyperparameters using metaheuristic strategies. More recently, attention-based and Transformer-inspired architectures have been explored to capture long-range temporal dependencies. Ran et al.[16] integrated empirical

mode decomposition with a Transformer structure to enhance temporal feature extraction, whereas Jiang et al.[17] and Li et al.[18] proposed extended attention frameworks to improve prediction performance under complex load patterns. In addition, hybrid DL models integrating multiple paradigms have also been investigated. Chen et al.[19] developed a Transformer–LSTM framework for industrial STLF, and Guo et al.[20] proposed a CNN–LSTM with multi-modal attention (CNN–LSTM–MMA) model to fuse heterogeneous inputs such as load demand and meteorological variables.

Despite these advances, increasing model depth and structural complexity often introduce challenges such as gradient degradation, unstable convergence, and performance saturation when modeling highly nonlinear load dynamics. These limitations highlight the need for learning frameworks that can support deep architectures while maintaining stable training behavior. DRNs, which incorporate identity-based skip connections to facilitate gradient propagation, provide an effective mechanism to alleviate these issues. By enabling deeper feature representation while preserving convergence stability, DRNs offer a promising foundation for modeling complex temporal variations in STLF tasks. Compared with conventional DL architectures, residual learning allows the network to focus on incremental feature refinement rather than complete transformation, thereby improving training efficiency and robustness under varying load conditions.

Although CNN-, recurrent-, attention-, and hybrid-based forecasting models have significantly advanced STLF research, most existing studies primarily focus on architectural innovation or feature engineering, while training strategies—particularly optimization methods—remain comparatively underexplored. Many studies tend to rely on default optimization settings during model training, with limited systematic investigation into how different gradient-based optimization strategies influence the behavior of DL models. This observation suggests that optimization strategies remain an important yet insufficiently studied aspect in DL-based STLF, providing motivation for subsequent research within residual learning frameworks.

DRN-based methods for short-term load forecasting

DRNs have demonstrated distinct advantages in STLF by effectively modeling highly nonlinear load dynamics while maintaining stable training behavior. Through identity-based residual connections, DRNs alleviate gradient degradation and enable deeper network architectures with improved convergence stability and feature representation capability. As a result, DRN-based models have emerged as a promising framework for enhancing the accuracy and robustness of STLF systems.

Early DRN-related studies mainly focused on validating the feasibility of residual learning in load forecasting tasks. Chen et al.[22] proposed a DRN-based forecasting framework that integrates residual blocks with ensemble learning strategies, demonstrating improved generalization performance compared with conventional DL baselines. Similarly, Kondaiah et al.[23] introduced a task-oriented DRN architecture tailored for STLF and confirmed that residual designs can effectively capture nonlinear load behaviors across multiple datasets. These studies established the foundation for applying residual learning mechanisms to load forecasting problems.

Subsequent research has explored various directions to enhance DRN performance, including ensemble strategies, structural refinements, and hybrid modeling approaches. Xu et al.[24] proposed an Ensemble Residual Network (ERN) with snapshot ensemble learning to improve generalization without training multiple independent models, while Chen et al.[25] introduced a ResNet-based ensemble framework incorporating multi-scale feature extraction to mitigate overfitting under varying load conditions. In parallel, data-driven enhancements have also been investigated. Kondaiah et al.[26] presented a Deep-ResNet framework emphasizing adaptive feature selection to improve learning stability across different operating environments. Structural refinements have further been explored to enhance representation capability, such as the incorporation of inception-inspired modules by Ding et al.[27] and redesigned convolutional residual blocks by Sheng et al.[28] for improved local pattern extraction.

In addition to architectural refinements, hybrid DRN frameworks combining residual learning with recurrent or attention-based mechanisms have been proposed to strengthen temporal dependency modeling. Tian et al.[29] introduced a ResNetPlus–LSTM framework to jointly exploit deep residual feature extraction and sequential modeling, while Li et al.[30] developed a ResNet–LSTM–Attention model to enhance temporal feature selection. More recently, Sheng et al.[31] proposed a Residual LSTM Plus framework that tightly integrates residual and recurrent components to improve forecasting robustness under complex load conditions. Although these hybrid designs achieve improved predictive performance, the increased architectural complexity often introduces additional computational overhead and challenges in model configuration.

To address the limitation of relying solely on temperature variables in earlier DRN-based studies, the PCA-DRN framework was subsequently proposed to incorporate multiple meteorological inputs through principal component analysis (PCA)[32]. By reducing dimensionality and mitigating redundancy among weather variables, PCA-DRN extends the original DRN architecture while preserving its residual learning structure. This approach enables richer environmental information to be integrated into forecasting models without significantly increasing architectural complexity.

Despite the extensive exploration of architectural design, ensemble strategies, and feature engineering in DRN-based STLF research, optimization strategies have received comparatively limited attention. While previous studies have focused on improving network structures or input representations, the influence of training algorithms on convergence behavior and forecasting performance has rarely been examined in a systematic manner. In particular, most existing DRN-based STLF studies[22–32] implicitly adopt the Adam optimizer as a default configuration, without comprehensive comparison against alternative gradient-based optimization algorithms. Given the depth and structural characteristics of residual networks, optimizer selection may substantially affect convergence stability, training dynamics, and predictive accuracy. Therefore, a systematic investigation of gradient-based optimization strategies within DRN frameworks is essential to better understand their role in STLF performance, which forms the primary motivation of this study.

Comparative analysis of gradient-based optimization algorithms in deep learning

In DL, the choice of an optimization algorithm plays a role as critical as network architecture and loss function design, as it directly affects training dynamics, convergence speed, and generalization performance[33]. This issue is particularly pronounced in time-series forecasting tasks such as STLF, where load data exhibit strong seasonality, non-stationarity, and volatility[34]. Under such conditions, inappropriate optimizer selection may result in unstable training behavior, slow convergence, or suboptimal forecasting accuracy, even when advanced model architectures are employed.

Empirical evidence from time-series forecasting studies indicates that optimizer choice and hyperparameter configuration can significantly influence convergence characteristics and prediction performance, including within residual-based architectures[33]. Moreover, existing research suggests that optimizer effectiveness is often task- and data-dependent, implying that conclusions drawn from one application domain may not be directly transferable to others[35]. These observations highlight the necessity of systematic and domain-specific investigations of gradient-based optimization algorithms rather than relying on default optimizer settings.

Several comparative studies have examined the performance of gradient descent–based optimization algorithms across different DL applications. Kondaiah et al.[36] conducted a comprehensive evaluation of nine optimizers—Stochastic Gradient Descent (SGD), Stochastic Gradient Descent with Momentum (SGD-Momentum), Nesterov Accelerated Gradient (NAG), Adaptive Gradient (AdaGrad), Adaptive Delta (AdaDelta), Root Mean Square Propagation (RMSProp), Adam, Adamax (an Adam variant based on the infinity norm), and Nesterov-accelerated Adaptive Moment Estimation (Nadam)—within CNN frameworks. While this study provided valuable insights into optimizer behavior for image classification tasks, its conclusions were limited to CNN architectures and did not address time-series forecasting problems.

Mustapha et al.[37] extended optimizer comparisons to medical image analysis by evaluating nine commonly used optimizers on ophthalmology datasets. Their results showed that AdaGrad achieved superior convergence accuracy, whereas AdaDelta performed the worst. However, the relatively small dataset size and application-specific nature of the task limited the generalizability of their conclusions. Murthy et al.[38] focused on agricultural image classification and compared three optimizers—SGD, RMSProp, and Adam—demonstrating that Adam significantly outperformed the others in classification accuracy. Although practically relevant, this study considered only a limited subset of optimization algorithms. Similarly, Sembiring et al.[39] investigated seven optimizers—SGD, SGD-Momentum, NAG, AdaGrad, AdaDelta, RMSProp, and Adam—on a body motion time-series dataset, revealing noticeable performance differences in sequential feature learning. Nevertheless, their evaluation primarily emphasized prediction accuracy, with limited discussion of convergence stability and generalization behavior.

Despite the efforts reported in the aforementioned studies, DRN-based STLF research has predominantly adopted Adam as the default optimizer, without systematically examining the influence of alternative gradient-based optimization strategies. Consequently, the potential impact of optimizer selection on training stability, convergence behavior, and forecasting accuracy within deep residual architectures remains insufficiently understood. This limitation is particularly critical given the depth and complexity of DRN models and the challenging characteristics of load time-series data.

Based on the optimization strategies reported in the aforementioned studies, gradient-based optimization algorithms can be broadly categorized into two major groups: classical stochastic gradient descent–based methods and adaptive gradient-based methods, which represent the two dominant optimization paradigms in DL. The classical SGD-based methods include SGD, SGD-Momentum, and NAG, which rely on first-order gradient information and momentum mechanisms to accelerate convergence. In contrast, adaptive gradient-based methods dynamically adjust learning rates based on historical gradient statistics and moment estimation, including AdaGrad, AdaDelta, RMSProp, Adam, Adamax, and Nadam.

Building upon adaptive gradient-based optimization methods, several improved variants of Adam have been proposed to address its known limitations. Adaptive Moment Estimation with Maximum Second-Moment (AMSGrad) employs a maximum second-moment estimator to modify the variance estimation strategy, thereby ensuring more stable convergence guarantees[40]. Adam with Decoupled Weight Decay (AdamW) separates weight decay from the learning-rate update, improving regularization effectiveness and generalization performance[41]. Rectified Adam (RAdam) introduces a variance rectification mechanism to alleviate instability caused by adaptive learning-rate variance during the early stages of training, leading to more reliable convergence behavior[42]. Adaptive Belief Optimizer (AdaBelief) further refines the adaptive update by measuring the "belief" in the observed gradients, enhancing generalization while maintaining fast convergence characteristics[43].

Based on the existing literature, although gradient-based optimization algorithms play a critical role in DL training, their potential impact on improving forecasting accuracy within DRN-based STLF frameworks has not been systematically investigated. In particular, existing studies predominantly adopt Adam as the default optimizer, while limited attention has been paid to the comparative performance of other gradient-based optimization strategies in deep residual architectures. In this context, it is necessary to conduct a systematic comparison of the aforementioned eleven representative gradient-based optimizers within a unified DRN framework, with the aim of identifying optimization strategies that can effectively enhance training efficiency and further improve forecasting accuracy, thereby providing a basis for performance optimization of DRN models in STLF.

Research methodology

Research data

This study utilizes two real-world load datasets—the dataset from the New England Independent System Operator (ISO-NE) and the Malaysia Petaling Jaya (MyPJ) dataset—to provide a comparative analysis of STLF under different climatic conditions and demand characteristics. The ISO-NE dataset contains hourly load and

temperature records from March 2003 to December 2006, representing a temperate climate with pronounced interannual and seasonal variability. The dataset covers six states in the New England power grid of the United States, including Connecticut, Maine, Massachusetts, New Hampshire, Rhode Island, and Vermont, and employs regional hourly observed temperature as input rather than forecasted values. In contrast, the Malaysia Petaling Jaya (MyPJ) dataset provides a reliable basis for examining electricity demand patterns in tropical climates. It consists of nationwide hourly load data obtained from the Malaysian Grid System Operator under Tenaga Nasional Berhad, together with daily meteorological observations collected in the Petaling Jaya region by the Malaysian Meteorological Department over the period from January 2020 to December 2022. The meteorological variables include rainfall, mean temperature, minimum temperature, maximum temperature, mean wind speed, maximum wind speed, and maximum wind direction. The selected region exhibits a typical tropical climate characterized by relatively limited seasonal variation, consistently high temperatures, and regular rainfall throughout the year, although discernible fluctuations in temperature and precipitation still influence electricity consumption patterns.

In real-world power system datasets, data-quality issues such as noise contamination, missing values, incomplete records, and unprocessed formats are commonly observed[44]. In terms of data preprocessing, the ISO-NE dataset was directly adopted as a benchmark since it had already undergone preliminary cleaning and standardization by the data provider. In contrast, missing values were identified in the raw MyPJ dataset during collection. To ensure stable model training and consistent performance evaluation, missing values in both load and meteorological variables were filled using linear interpolation, thereby establishing a complete and standardized baseline dataset. This preprocessing step aimed to provide a clean and consistent reference environment, ensuring the reliability and reproducibility of subsequent analyses.

Figure 1 illustrates the hourly load profiles of both datasets over the entire study period. The ISO-NE load values generally range from approximately 10,000 to 27,500 megawatts (MW) and exhibit clear seasonal and long-term fluctuations, whereas the MyPJ load values mainly fall between about 10,000 and 18,000 MW, reflecting relatively stable demand characteristics typical of tropical regions. After preprocessing, both datasets were divided into training and testing subsets. All input variables were normalized using statistical parameters derived solely from the training data to prevent information leakage, maintain consistent scaling, and improve model stability during both training and evaluation phases.

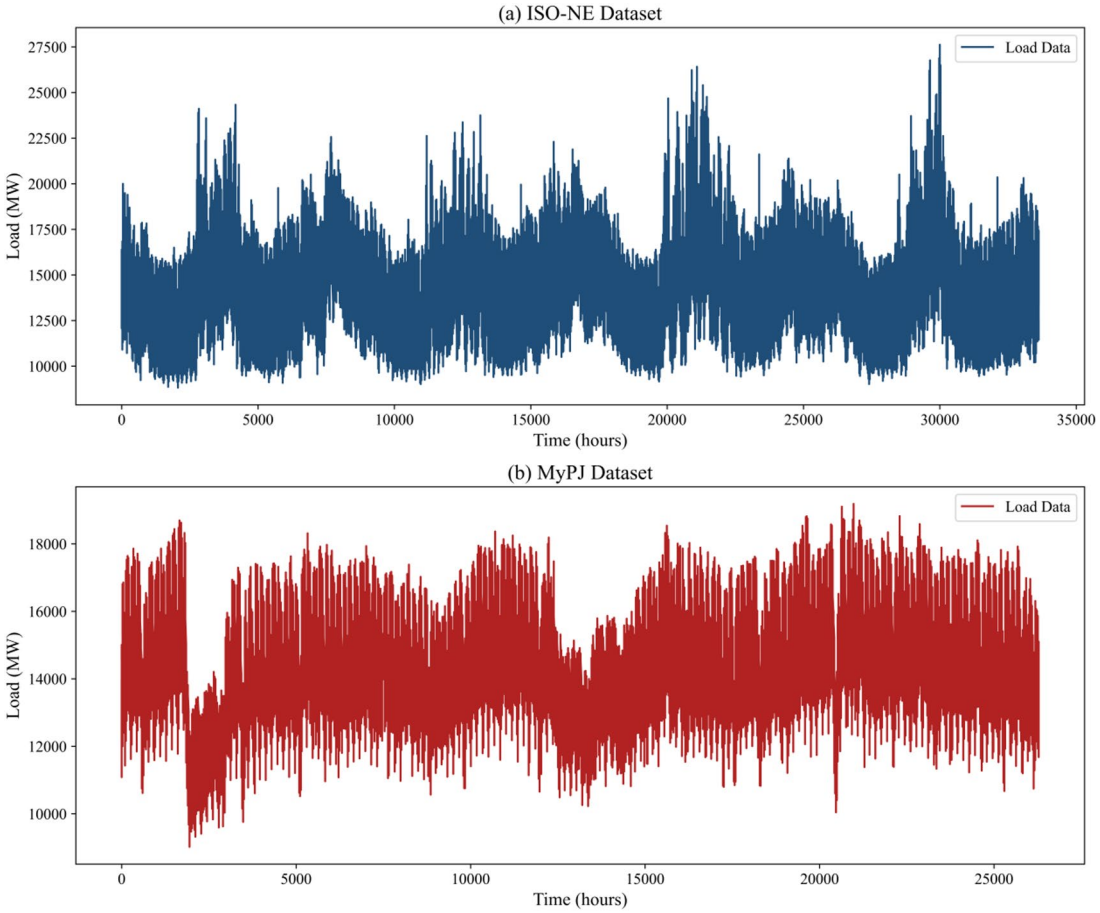

Fig. 1. Hourly load profiles of the ISO-NE and MyPJ datasets.

Architectures of DRN and PCA-DRN

To provide a clear architectural context for DRN-based STLF models, the original DRN is widely regarded as a representative framework due to its extensive adoption in existing studies. Meanwhile, PCA-DRN has been proposed as an extension of the DRN architecture to incorporate richer meteorological information through PCA, while preserving the core residual learning structure. By maintaining an identical residual network architecture, PCA-DRN enables the integration of multidimensional weather information without altering the fundamental design of the original DRN. The detailed architectures of both models are described as follows.

The architectures of the original DRN[22] and its PCA–based extension, referred to as PCA-DRN[32], are illustrated in Figs. 2 and 3, respectively. Both frameworks consist of two main components: a basic structure and a ResNetPlus module. In the original DRN formulation for STLF, weather information is incorporated exclusively through temperature variables, which are combined with historical load and time-related features to characterize demand variations under different temporal conditions. The basic structure processes these inputs to generate preliminary load representations, which are subsequently refined by the ResNetPlus module—an enhanced variant of the conventional ResNet architecture—through deep residual learning to strengthen feature representation and improve 24-hour-ahead forecasting accuracy.

Despite its effectiveness, the original DRN framework considers temperature as the sole meteorological input, whereas practical STLF problems are influenced by multiple weather variables that often exhibit strong correlations and redundancy. To address this limitation, PCA-DRN extends the original DRN framework by incorporating multiple meteorological variables and applying PCA during the data preprocessing and feature construction stages, thereby enabling dimensionality reduction and redundancy elimination while preserving the original two-component architecture. In both DRN and PCA-DRN, day-ahead STLF is realized through a sequence of hour-specific sub-models, each designed to predict the load of a particular future hour, with the outputs of preceding sub-models progressively fed back into the network to capture short-term inter-hour dependencies. The resulting 24 hourly predictions are subsequently jointly optimized by the ResNetPlus module, which refines the overall forecast and produces the final day-ahead load profile. For both models, the Scaled Exponential Linear Unit (SELU) is employed as the activation function for all hidden layers with the LeCun normal initializer, while a linear activation function is used in the output layer to generate the final load prediction.

Within the original DRN framework, as well as its PCA-DRN extension, model optimization is driven by a unified training objective that balances prediction accuracy and the structural consistency of daily load profiles. Given that DRN-based forecasting aims not only to minimize point-wise errors but also to preserve the realistic range of predicted load curves, the overall loss function is formulated as the combination of two complementary components. As shown in Eq. (1), the total loss is defined as the sum of an error-based term $\mathrm{Loss_E}$ and a rangeaware penalty term $\mathrm{Loss_R}$. Specifically, $\mathrm{Loss_E}$, expressed in Eq. (2), computes the mean absolute percentage error (MAPE) by evaluating the relative prediction error between the predicted normalized load $\mathrm{y(j,h)}$ and the actual normalized load $\mathrm{y(j,h)}$ for the h-th hour of the j-th day, where H (set to 24) denotes the number of hourly load values per day and Num represents the total number of samples. To further constrain the predicted daily load curve within a physically reasonable range, Eq. (3) introduces $\mathrm{Loss_R}$, which penalizes overestimation of daily peak loads and underestimation of trough values by comparing the predicted and actual maximum and minimum loads. By explicitly enforcing range consistency, this penalty term complements the MAPE-based error term and enhances the robustness of learning within the DRN-based framework.

$$\mathrm{Loss} = \mathrm{Loss_E} + \mathrm{Loss_R} \tag{1}$$

$$\mathrm{Loss_E} = \frac{1}{\mathrm{NumH}} \sum_{\mathrm{j}=1}^{\mathrm{N}} \sum_{\mathrm{h}=1}^{\mathrm{H}} \frac{\left|\hat{\mathrm{y}}_{(\mathrm{j,h})} - \mathrm{y}_{(\mathrm{j,h})}\right|}{\mathrm{y}_{(\mathrm{j,h})}} \tag{2}$$

$$\begin{aligned} \mathrm{Loss_R} = \frac{1}{2\mathrm{Num}} \sum_{\mathrm{j}=1}^{\mathrm{Num}} & \max\left(0, \max_{\mathrm{h}} \hat{\mathrm{y}}_{(\mathrm{j,h})} - \max_{\mathrm{h}} \mathrm{y}_{(\mathrm{j,h})}\right) \\ & + \max\left(0, \min_{\mathrm{h}} \mathrm{y}_{(\mathrm{j,h})} - \min_{\mathrm{h}} \hat{\mathrm{y}}_{(\mathrm{j,h})}\right) \end{aligned} \tag{2}$$

Architectures of DRN on ISO-NE dataset

As illustrated in Fig. 2, the architecture of the DRN framework on the ISO-NE dataset consists of two main components, namely the basic structure and the ResNetPlus module. As the first component of the original DRN, the basic structure is implemented as a feedforward neural network composed of multiple interconnected fully connected (FC) layers and is responsible for generating the initial load forecasts for the subsequent 24-hour horizon. Within this topology, each FC layer contains ten hidden neurons and is associated with the feature groups $[\mathrm{L_h^{day}}, \mathrm{T_h^{day}}]$, $[\mathrm{L_h^{week}}, \mathrm{T_h^{week}}]$, $[\mathrm{L_h^{month}}, \mathrm{T_h^{month}}]$ and $\mathrm{L_h^{hour}}$. In addition, intermediate FC layers linked to categorical temporal information-namely weekday and seasonal indicators represented by the one-hot encoded variables W and S-are configured with five hidden neurons each, while the FC1, FC2, and the layer preceding $\mathrm{L_h}$ also employ ten hidden neurons. Activation functions are applied to all layers except the output layer. In this basic structure, $\mathrm{L_h^{month}}$ denotes the load values corresponding to the same hour from one, two, and three months prior to the target day, whereas $\mathrm{L_h^{week}}$ captures the load values for the same hour over the preceding one to eight weeks. The variable $\mathrm{L_h^{hour}}$ records load observations for the same hour during the previous 24 h, while $\mathrm{L_h^{day}}$ represents the load values for the same hour on each day of the previous week. Correspondingly, the temperature variables $\mathrm{T_h^{month}}$, $\mathrm{T_h^{week}}$ and $\mathrm{T_h^{day}}$ are aligned with $\mathrm{L_h^{month}}$, $\mathrm{L_h^{week}}$ and $\mathrm{L_h^{day}}$, respectively, where

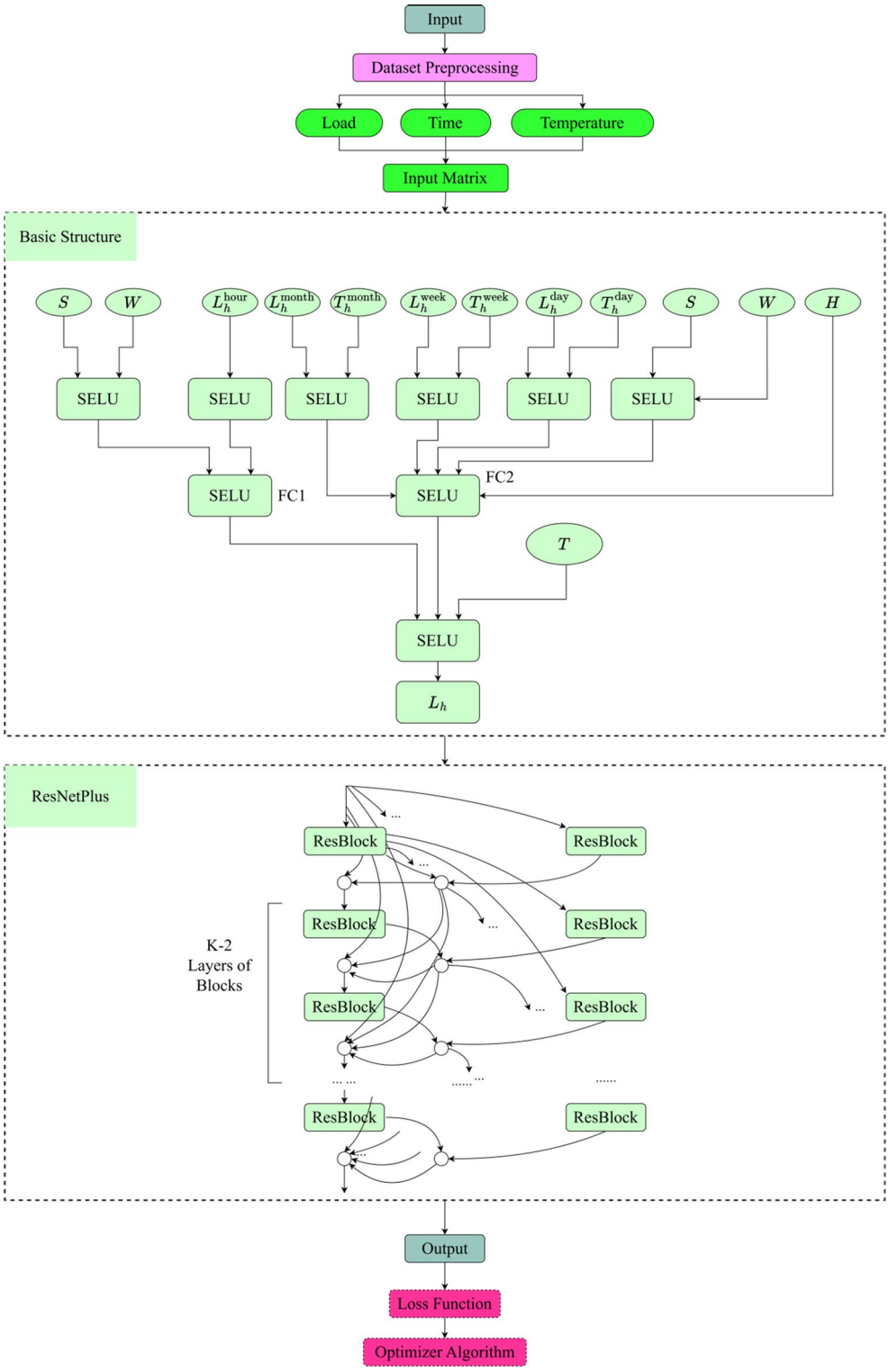

Fig. 2. Architecture of the DRN framework on the ISO-NE dataset[22].

T_h denotes the actual observed temperature for the following day. Moreover, the categorical inputs S, W, and H represent seasonal condition, weekday, and holiday status via one-hot encoding. The output produced by this fundamental module is subsequently passed to the second component of the DRN, where it is further refined to improve overall forecasting accuracy. Specifically, S corresponds to the four seasons—spring, summer, autumn, and winter—whereas H includes major public holidays such as Christmas and Independence Day.

As the second component of the original DRN architecture, ResNetPlus constitutes an enhanced refinement of a residual-learning design that introduces targeted architectural optimizations while maintaining efficient

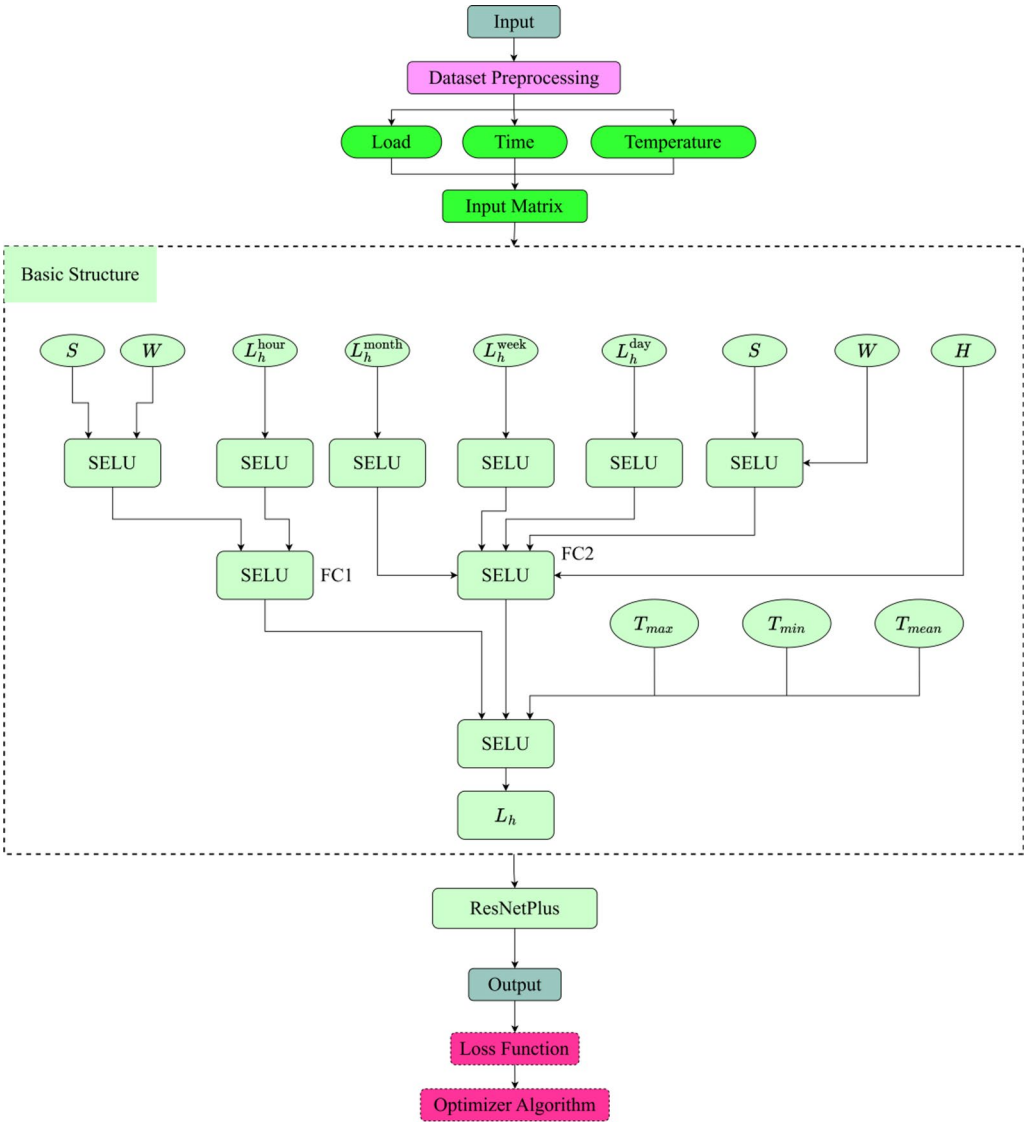

Fig. 3. Architecture of the DRN framework on the MyPJ dataset.

Item	Setting
Residual block structure	Two layers per block (SELU nonlinear layer + Linear projection layer)
Hidden neurons (SELU layer)	20 neurons
Number of Internal Residual Blocks	4 blocks aggregated via Add operation
Hierarchical Levels	10 stacked levels
Shortcut Connections	Identity shortcut connections enabled
Activation Function	SELU (hidden layers); Linear (output layer)
Kernel Initializer	LeCun Normal
Normalization Layers	Not applied
Dropout / Alpha Dropout	Not applied

Table 1. Key configuration of the ResNetPlus module.

gradient propagation. The key architectural settings of the ResNetPlus module are summarized in Table 1 for clarity. It is constructed from a sequence of residual blocks, each consisting of two layers: a nonlinear hidden layer with 20 neurons activated by the SELU, consistent with the activation used in the basic structure, followed by a linear transformation layer that aligns feature dimensions to enable residual addition. The residual structure comprises four internal residual blocks, where each block independently performs nonlinear

feature transformation and linear projection on the input representations. The outputs of these four blocks are subsequently aggregated through an Add operation to form residual connections, thereby establishing stable gradient propagation paths within the deep network. Based on this design, the four-block structure is repeatedly stacked across ten hierarchical levels to construct a structurally deep and expressive architecture. A defining characteristic of ResNetPlus is the inclusion of shortcut connections, which allow the output of a preceding block to be directly propagated forward, simplifying deep network construction and improving computational efficiency. While retaining the residual-learning philosophy, ResNetPlus reorganizes the block composition to more effectively leverage the advantages of deep residual modeling within the DRN framework.

Architectures of DRN and PCA-DRN on MyPJ dataset

For the MyPJ dataset, the architecture of the DRN framework is illustrated in Fig. 3 and follows the same residual learning structure as the original DRN, with modifications mainly applied to the meteorological inputs. The historical load features retain the same multi-scale temporal construction, while the meteorological variables consist of daily statistical temperature features, including the daily mean, maximum, and minimum temperatures ($\mathrm{T_{mean}}, \mathrm{T_{max}}, \mathrm{T_{min}}$), which are concatenated and processed through an additional FC layer with ten hidden neurons. The categorical variables S and W are encoded through dedicated FC layers with five hidden neurons each, while the intermediate layers (FC1 and FC2) and the layer preceding $\mathrm{L_h}$ maintain ten hidden neurons to ensure sufficient representation capacity. The categorical variables are incorporated as part of the model inputs, where S represents the two climatic seasons (rainy and dry seasons) and H includes major public holidays such as Eid al-Fitr and Malaysia Independence Day. The aggregated output of the basic structure, denoted as $\mathrm{L_h}$, is subsequently fed into the ResNetPlus module. Notably, the ResNetPlus configuration on the MyPJ dataset remains identical to that of the original DRN, including the residual block design, activation settings, and network depth.

To overcome the limitations associated with relying on a restricted set of meteorological inputs, the PCA-DRN framework, as illustrated in Fig. 4, incorporates multiple weather variables through a PCA-based preprocessing strategy applied prior to model training. In this framework, meteorological features including temperature, wind-related variables, and rainfall measurements are first integrated and transformed using PCA. The PCA process reduces dimensionality and eliminates redundancy while retaining principal components that collectively explain approximately 90% of the cumulative variance, which are subsequently used as weather-related inputs to the model's basic structure. Apart from replacing the raw meteorological variables with PCA-transformed components, the remaining architectural elements remain consistent with the original DRN. Specifically, the FC layers corresponding to $\mathrm{L_h^{day}}$, $\mathrm{L_h^{week}}$, $\mathrm{L_h^{month}}$ and $\mathrm{L_h^{hour}}$, together with the one-hot encoded categorical variables S and W, are preserved, and the aggregated output $\mathrm{L_h}$ is passed to the ResNetPlus module following the same residual learning process. To ensure consistent architectural conditions for performance comparison, all FC layer configurations-including neuron numbers, connectivity patterns, and activation settings -as well as the ResNetPlus component, are kept identical to those in the DRN framework.

Proposed methodology

This study adopts a unified methodological framework to systematically evaluate the influence of gradient-based optimization algorithms on DRN-based STLF. Two forecasting configurations are investigated within the same experimental pipeline, namely the original DRN and its meteorologically enhanced variant, PCA-DRN. The overall workflow integrating both configurations is illustrated in Fig. 5.

As shown in Fig. 4, the forecasting process begins with data input and dataset preprocessing. The raw inputs include historical load data, time-related variables, and meteorological features. Dataset preprocessing involves missing-value handling, normalization based on training-set statistics, and feature alignment, ensuring numerical stability while preventing information leakage during model training and evaluation. After preprocessing, the constructed features are fed into the DRN architecture, which serves as the core forecasting engine for generating day-ahead hourly load predictions. Through deep residual learning, the model captures complex nonlinear relationships between multi-scale historical information and future electricity demand.

Within this unified framework, two meteorological representation strategies are investigated. The baseline DRN utilizes temperature-related variables together with historical load and temporal inputs. In contrast, the PCA-DRN configuration incorporates richer meteorological information by applying principal component analysis (PCA) to multiple weather variables prior to model training. PCA reduces dimensionality and eliminates redundancy while preserving the dominant variance structure of the original meteorological data. The resulting PCA-transformed components replace the raw weather inputs and are combined with historical load and temporal features before entering the DRN model. Apart from this transformation of meteorological representation, the remaining elements of the forecasting pipeline—including dataset preprocessing, model architecture, and training strategy—remain identical, ensuring that any observed performance differences primarily reflect feature representation rather than structural variation.

For both DRN and PCA-DRN configurations, model training is guided by a predefined loss function that measures the discrepancy between predicted and observed load values and serves as the optimization objective for parameter updates. Based on the computed loss, network parameters are updated using gradient-based optimization algorithms. To systematically investigate the influence of optimizer selection on training dynamics and forecasting performance, a comprehensive set of representative optimizers is evaluated. These include classical stochastic gradient descent–based methods (SGD, SGD-Momentum, and NAG) and adaptive gradient-based methods (AdaGrad, AdaDelta, RMSProp, Adam, Adamax, Nadam, AdamW, AMSGrad, RAdam, and AdaBelief).

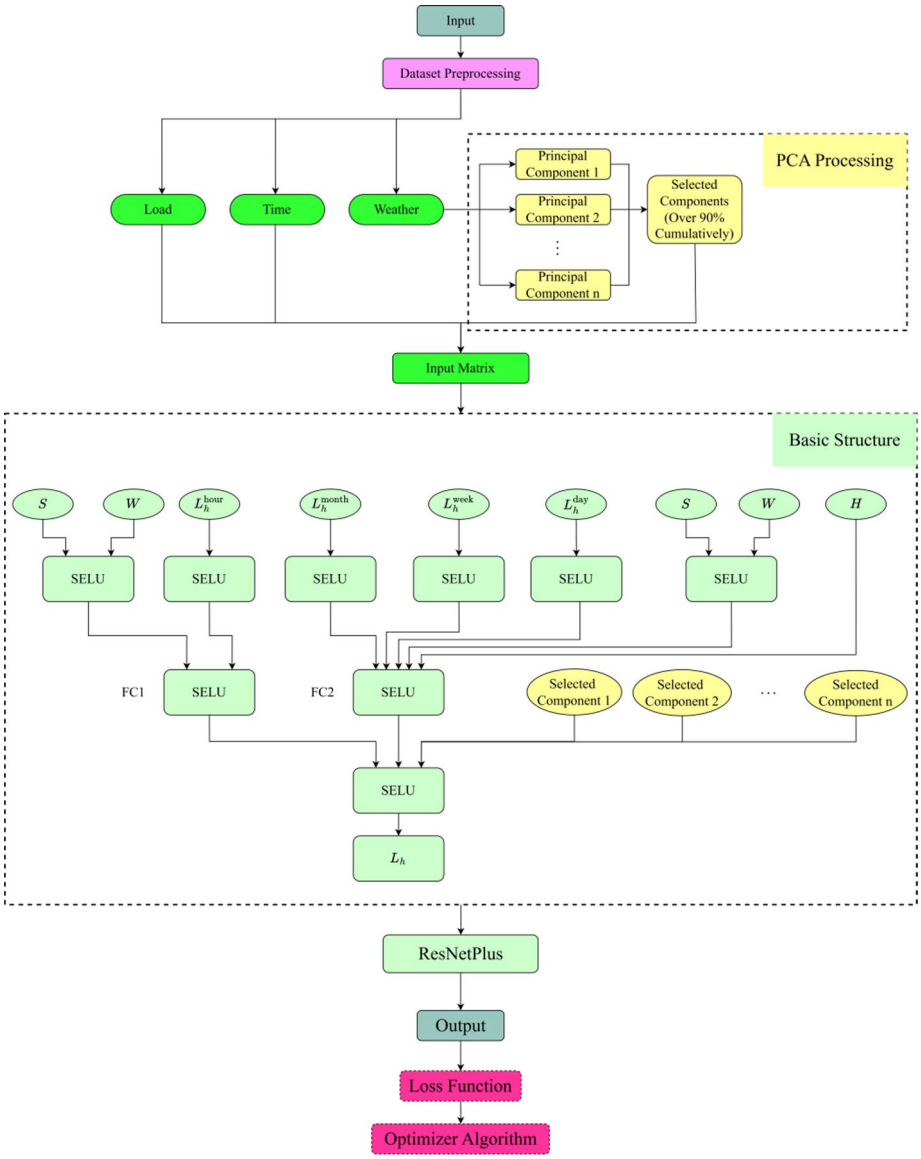

Fig. 4. Architecture of the PCA-DRN framework on the MyPJ dataset.

Importantly, the same loss formulation, training configuration, and optimizer set are consistently applied to both DRN and PCA-DRN models. By maintaining identical optimization conditions across different forecasting configurations, the proposed methodology isolates the effect of optimizer selection from architectural and feature-representation factors. This unified experimental design enables a fair and systematic evaluation of how different gradient-based optimization algorithms influence convergence behavior, training stability, and forecasting accuracy in DRN-based STLF.

Experimental design

This study employs real-world electricity load datasets from both ISO-NE and MyPJ. For the ISO-NE dataset, observations collected from March 2003 to December 2005 were used for model training, and data from the year 2006 were reserved for testing, resulting in 24,888 training samples and 8760 testing samples. For the MyPJ dataset, observations collected from January 2020 to December 2021 were used for model training, and data from the year 2022 were reserved for testing, resulting in 17,544 training samples and 8760 testing samples. The datasets offer a robust basis for assessing forecasting performance under temperate and tropical climatic conditions.

To provide representative benchmarking and contextual comparison, several widely adopted DL models reported in the STLF literature are selected as reference baselines. These include CNN, LSTM, and Transformer models, as well as two representative hybrid architectures—Transformer–LSTM[19] and CNN–LSTM–MMA[20]. The benchmarking comparison with these baseline models is conducted on the ISO-NE dataset, enabling a direct architectural comparison between the proposed DRN framework and widely adopted DL models under a controlled temperate-climate scenario that follows commonly used experimental settings in existing studies.

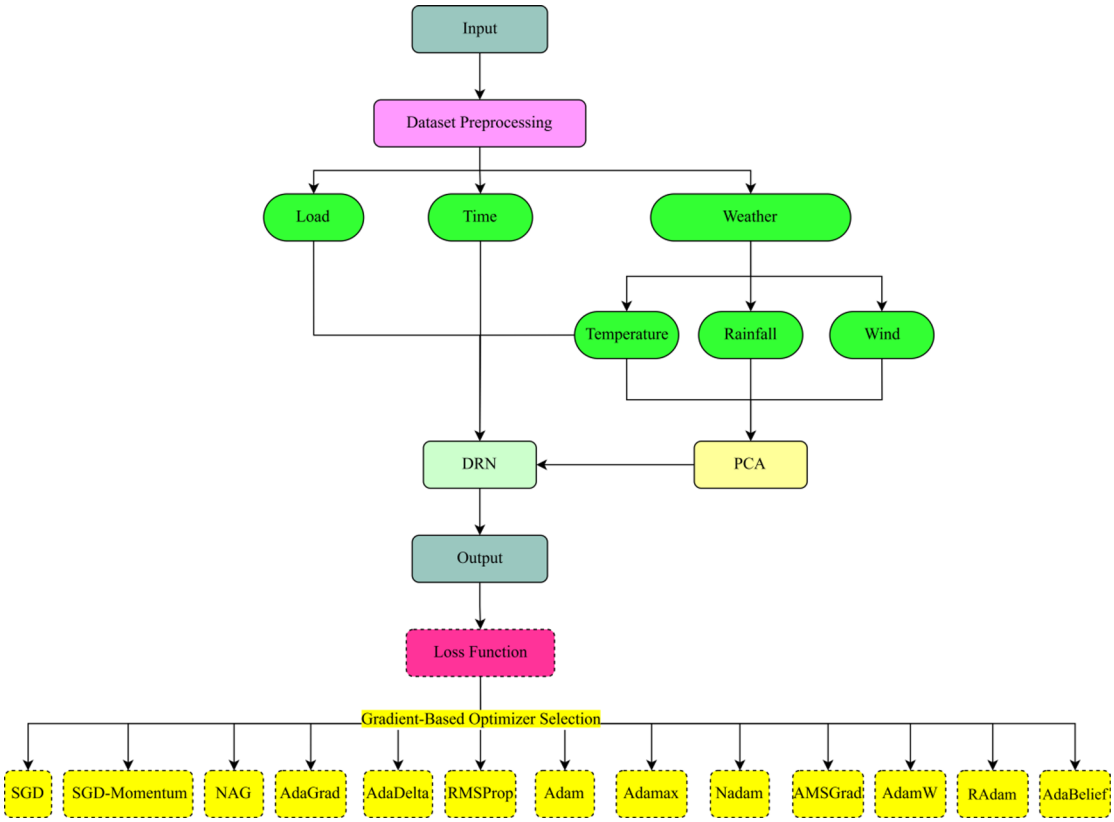

Fig. 5. Unified framework for evaluating gradient-based optimization algorithms in DRN-based STLF.

The inclusion of these baseline models aims to evaluate the relative effectiveness of the DRN architecture against conventional DL approaches, rather than to conduct an exhaustive comparison across all model families. Consequently, these baselines serve as contextual references to demonstrate the architectural advantages of DRN, while the primary focus of this study remains the investigation of optimization strategies within residual-based forecasting frameworks.

To ensure reproducibility and methodological transparency, the baseline models are implemented using standard configurations consistent with their original formulations. The CNN baseline adopts a one-dimensional convolution (Conv1D) architecture with 64 filters and a kernel size of 3, followed by a Rectified Linear Unit (ReLU) activation function. The LSTM model employs 64 recurrent units, using the hyperbolic tangent (Tanh) activation for the cell state and logistic sigmoid functions for the gating mechanisms. A canonical Transformer configuration is adopted, comprising a single encoder layer with eight attention heads of 64 dimensions each, a 2048-dimensional position-wise feed-forward network, a 64-dimensional embedding space, positional encoding, and a dropout rate of 0.1. For the Transformer–LSTM and CNN–LSTM–MMA models, architectural settings are retained as reported in their original studies to maintain consistency and fairness in comparison. To avoid confounding effects arising from optimization strategies, all baseline models are trained using the Adam optimizer with default parameter settings. For a fair and consistent comparison, the DRN model are also trained using the Adam optimizer under the same settings when compared with these baseline models. As a result, comparisons involving these baseline models focus exclusively on architectural characteristics and forecasting capability, rather than on optimizer-related effects. These baseline models are therefore not included in the subsequent investigation of gradient-based optimization algorithms, which constitutes the primary focus of this study.

To systematically examine the influence of optimizer selection on training dynamics and forecasting performance, thirteen representative gradient-based optimization algorithms are evaluated exclusively within the DRN and PCA-DRN frameworks. These include classical stochastic gradient descent–based methods (SGD, SGD-Momentum, and NAG) and adaptive gradient-based methods (AdaGrad, AdaDelta, RMSProp, Adam, Adamax, Nadam, AdamW, AMSGrad, RAdam, and AdaBelief). To ensure methodological fairness and reproducibility, all optimizers are evaluated under a unified hyperparameter configuration protocol. A consistent learning-rate search strategy is first conducted on the ISO-NE dataset using the DRN framework, where each optimizer is tested under the same set of learning rates {0.0001, 0.0003, 0.001, 0.003, 0.01}. This range covers commonly adopted learning-rate magnitudes on a logarithmic scale, allowing both conservative and aggressive update behaviors to be evaluated without introducing optimizer-specific bias. For each optimizer, the best-performing learning rate identified from the ISO-NE evaluation stage is selected for subsequent comparison. To avoid dataset-specific hyperparameter tuning, the selected configurations are directly transferred to the

MyPJ dataset, where both DRN and PCA-DRN models are evaluated under identical settings without additional adjustment. Apart from the learning rate, all remaining optimizer-specific hyperparameters, including momentum or β parameters, weight decay, and internal scheduling mechanisms, are retained at their default settings as defined in their original implementations. This unified design ensures that the search space and computational budget remain consistent across optimizers, thereby enabling a fair comparison focused on intrinsic optimization behavior rather than extensive optimizer-specific tuning.

Two rounds of short-term fine-tuning of 50 epochs each followed the first 600 epochs of normal training, for a total of 700 epochs of training. A snapshot ensemble was created during training by saving three model snapshots at the conclusion of each short-term phase. This method maintains model weights at various learning rate phases during training[22,45]. The method improves prediction stability and generalization capacity while successfully lowering the danger of overfitting associated with a single model by averaging predictions from these snapshots. In order to achieve similar resilience and stability in predicting performance, this ensemble technique provides a computationally effective substitute for carrying out several separate training trials[46].

To rigorously assess whether the observed performance gains of the enhanced model are statistically significant, a nonparametric Bootstrap resampling procedure with 10,000 iterations is employed. Unlike the paired Student's t-test, which requires the assumption that paired differences follow a normal distribution, the Bootstrap approach does not rely on any distributional assumptions, thereby providing a more robust and reliable framework for model performance comparison[47]. Statistical significance is determined using two complementary criteria: first, if the 95% confidence interval (CI) of the mean performance difference lies entirely above zero, the improvement is regarded as significant at the 95% confidence level; otherwise, if the interval includes zero, the difference is considered insignificant. Second, within the Bootstrap framework, a probability value (p-value) smaller than 0.05 likewise indicates statistical significance at the same confidence level. It should be emphasized that a reported value of $p \approx 0$ represents an extremely small probability (less than 0.0001) rather than an exact zero. MAPE is adopted as the evaluation metric due to its widespread use in STLF studies and its capability to provide an interpretable, scale-independent measure of relative forecasting error.

All experiments were conducted in a Python 3.8 environment with TensorFlow 2.10.0 and Keras 2.10.0 as the DL backends. The computations were performed on a Lenovo laptop equipped with an AMD Ryzen 7 6800 H CPU, 16 GB DDR5 4800 MHz RAM, and an NVIDIA GeForce RTX 3050 Ti Laptop GPU (4 GB).

Evaluation metrics

Consistent with previous DRN-based studies for STLF, multiple evaluation metrics are employed in this study to assess forecasting performance. Among these metrics, the MAPE is adopted as the primary and principal evaluation metric, owing to its intuitive interpretability and its broad and consistent adoption in the STLF literature. Other commonly reported error- and correlation-based metrics, including Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Normalized Mean Squared Error (NMSE), Correlation Coefficient (R), and Coefficient of Determination (R^2), are included to provide complementary performance information and to facilitate comparison with related studies, while MAPE serves as the main basis for model comparison and statistical analysis in this work. Depending on particular goals and dataset properties, many research may use distinct assessment measures. The related formulae are shown in Eqs. (4)–(10).

$$\mathrm{MAPE} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \left| \frac{\mathrm{y_i} - \hat{\mathrm{y}}_\mathrm{i}}{\mathrm{y_i}} \right| \times 100 \tag{4}$$

$$\mathrm{MAE} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i\text{-}1}}^{\mathrm{N}} |\mathrm{y_i} - \hat{\mathrm{y}}_\mathrm{i}| \tag{5}$$

$$\mathrm{MSE} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i\text{-}1}}^{\mathrm{N}} (\mathrm{y_i} - \hat{\mathrm{y}}_\mathrm{i})^2 \tag{6}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{\mathrm{N}} \sum_{\mathrm{i\text{-}1}}^{\mathrm{N}} (\mathrm{y_i} - \hat{\mathrm{y}}_\mathrm{i})^2} \tag{7}$$

$$\mathrm{NMSE} = \frac{\sum_{i-1}^{\mathrm{N}} (\mathrm{y_i} - \hat{\mathrm{y}}_\mathrm{i})^2}{\mathrm{N} \cdot \sigma_\mathrm{y}^2} \tag{8}$$

$$\mathrm{R} = \frac{\sum_{\mathrm{i\text{-}1}}^{\mathrm{N}} \left(\mathrm{y_i} - \overleftarrow{\mathrm{y}}\right) \left(\hat{\mathrm{y}}_\mathrm{i} - \bar{\hat{\mathrm{y}}}\right)}{\sqrt{\sum_{\mathrm{i\text{-}1}}^{\mathrm{N}} \left(\mathrm{y_i} - \overleftarrow{\mathrm{y}}\right)^2 \sum_{\mathrm{i\text{-}1}}^{\mathrm{N}} \left(\hat{\mathrm{y}}_\mathrm{i} - \bar{\hat{\mathrm{y}}}\right)^2}} \tag{9}$$

$$R^2 = 1 - \frac{\sum_{i-1}^{N} (y_i - \hat{y}_i)^2}{\sum_{i-1}^{N} \left(y_i - \overleftarrow{y}\right)^2} \tag{10}$$

These metrics' parameters are as follows: In the NMSE calculation, y_i denotes the actual value of the i-th sample, while $\hat{y}_i$ corresponds to its predicted counterpart.The symbols $\overleftarrow{y}$ and $\bar{\hat{y}}$ and σ_y^2 represent the mean of the actual and predicted series and the variance of the real values, respectively. N stands for the total number of samples. The parameter definitions for the remaining metrics follow the same notation. With the help of these characteristics, the metrics may evaluate performance in a more thorough way by taking into consideration factors like error magnitude, prediction precision, and the correlation between actual and anticipated values. While lower values for MAPE, MAE, MSE, RMSE and NMSE frequently imply fewer prediction errors and more generalization, higher R and R^2 values indicate better accuracy and model fitting.

Results and discussion

Baseline model comparison under adam optimization on the ISO-NE dataset

To establish an architectural reference prior to the investigation of optimization strategies, several representative DL models were evaluated on the ISO-NE dataset under a unified training configuration using the Adam optimizer. The purpose of this comparison is to provide contextual benchmarking for the DRN framework within commonly adopted DL approaches for STLF rather than to perform an exhaustive ranking across all model families.

Table 2 presents the forecasting performance of six representative models, including CNN, LSTM, Transformer, Transformer–LSTM, CNN–LSTM–MMA, and DRN. All models were trained and evaluated using identical preprocessing procedures, training–testing splits, and evaluation metrics to ensure methodological consistency. Conventional single-architecture models, such as CNN and LSTM, achieve moderate forecasting accuracy, with MAPE values of 0.024224 and 0.023277, respectively. The Transformer model further reduces the prediction error to a MAPE of 0.021925, indicating that attention-based mechanisms can capture long-range temporal dependencies more effectively than purely convolutional or recurrent structures.

Hybrid architectures that integrate multiple modeling paradigms demonstrate additional performance improvements. The Transformer–LSTM framework achieves a MAPE of 0.017280, suggesting that combining attention mechanisms with sequential modeling enhances temporal representation capability. Similarly, the CNN–LSTM–MMA model attains competitive performance with a MAPE of 0.019086, reflecting the benefit of multi-modal attention for integrating heterogeneous inputs.

Among the evaluated models, the DRN framework achieves the lowest forecasting error, with a MAPE of 0.017182 and strong correlation performance ($R = 0.989767$ and $R^2 = 0.979568$). Compared with conventional DL architectures, residual learning enables deeper hierarchical feature refinement while maintaining stable gradient propagation, which contributes to improved convergence behavior and enhanced predictive accuracy. Across additional evaluation metrics, including MAE, MSE, RMSE, and NMSE, similar trends can be observed, where architectures incorporating residual connections or hybrid feature extraction strategies consistently outperform single-structure baselines.

It should be emphasized that all baseline models in this comparison are trained exclusively using the Adam optimizer, ensuring that the observed differences primarily reflect architectural characteristics rather than optimization effects. These benchmarking results therefore provide a contextual reference for understanding the structural advantages of DRN-based forecasting before proceeding to a systematic investigation of gradient-based optimization strategies within the residual learning framework.

Optimizer hyperparameter sensitivity on the ISO-NE dataset

Learning-rate sensitivity analysis of gradient-based optimizers

To ensure a fair and systematic comparison among different gradient-based optimizers, a unified learning-rate sensitivity analysis was first conducted using the DRN framework on the ISO-NE dataset. The purpose of this analysis is not to identify the best forecasting performance directly, but rather to examine how different optimizers respond to variations in learning rate under identical training conditions and to determine suitable learning-rate configurations for subsequent experiments.

As shown in Table 3, all optimizers were evaluated using the same predefined learning-rate set {0.0001, 0.0003, 0.001, 0.003, 0.01}, which spans commonly adopted update magnitudes on a logarithmic scale. This

Model	MAPE	MAE	MSE	RMSE	NMSE	*R*	R^2
CNN	0.024224	0.015815	0.000595	0.024393	0.039482	0.980696	0.960518
LSTM	0.023277	0.015163	0.000542	0.023283	0.035968	0.982602	0.964032
Transformer	0.021925	0.014352	0.000517	0.022749	0.034337	0.982763	0.965663
Transformer-LSTM	0.017280	0.011185	0.000311	0.017624	0.020609	0.989707	0.979391
CNN-LSTM-MMA	0.019086	0.012157	0.000348	0.018645	0.023067	0.988523	0.976933
DRN	0.017182	0.011138	0.000308	0.017548	0.020432	0.989767	0.979568

Table 2. Benchmarking results of different models on the ISO-NE dataset.

Optimizer Algorithm	0.0001	0.0003	0.001	0.003	0.01
SGD	0.031156	0.027064	0.023103	**0.020318**	N/A
SGD-Momentum	0.030579	0.027416	**0.018898**	0.021973	N/A
NAG	0.019239	**0.016330**	0.018305	N/A	N/A
AdaGrad	0.088082	0.046231	0.029269	0.025170	**0.018937**
AdaDelta	0.482731	0.158406	0.081690	0.044761	**0.031948**
RMSProp	0.022493	**0.018494**	0.019661	0.021515	0.028248
Adam	0.023215	0.018719	0.017182	**0.016278**	0.028723
Adamax	0.025477	0.021979	0.017208	**0.016630**	0.016752
Nadam	0.023407	0.017791	0.017804	**0.017690**	0.028388
AMSGrad	0.022960	0.018580	0.017119	**0.015756**	0.017059
AdamW	0.025361	0.020483	0.019387	**0.016692**	0.028318
RAdam	0.023337	**0.017109**	0.017217	0.016737	0.029659
AdaBelief	0.023354	0.018117	**0.016336**	0.016391	0.054038

Table 3. Learning-rate sensitivity of gradient-based optimizers for DRN on the ISO-NE dataset (MAPE).

unified search space ensures methodological consistency across optimizers and prevents optimizer-specific bias during hyperparameter selection. Apart from the learning rate, all remaining optimizer parameters were retained at their default settings, allowing the analysis to focus specifically on learning-rate sensitivity while maintaining comparable training conditions. The notation "N/A" in Table 3 indicates that the corresponding training configuration failed to produce valid results due to unstable convergence behavior.

The results reveal noticeable differences in convergence behavior between classical stochastic gradient descent–based methods and adaptive gradient-based optimizers. Classical methods exhibit stronger dependence on learning-rate selection. For instance, SGD shows gradual performance improvement as the learning rate increases from 0.0001 to 0.003, suggesting slow convergence under smaller update magnitudes. SGD-Momentum achieves its best performance at a moderate learning rate of 0.001, while NAG demonstrates relatively stable performance at lower learning rates but becomes unstable when the learning rate increases, where valid results are no longer observed.

In contrast, adaptive gradient-based optimizers display more stable performance across multiple learning-rate values. RMSProp, Adam, Nadam, AMSGrad, AdamW, RAdam, and AdaBelief maintain relatively consistent MAPE levels within the range from 0.0003 to 0.003, indicating that adaptive moment estimation mechanisms can effectively regulate parameter updates under different step sizes. Among these methods, AMSGrad demonstrates the most consistent improvement trend as the learning rate increases and achieves the lowest MAPE value within the evaluated search space, indicating strong learning-rate robustness under the DRN framework. Adam also shows stable convergence behavior across multiple learning-rate settings, while AdaBelief achieves competitive performance at intermediate learning rates, suggesting balanced convergence dynamics. AdamW achieves its best performance at a learning rate of 0.003, reflecting improved convergence stability under moderate update magnitudes. Conversely, AdaGrad and AdaDelta demonstrate higher sensitivity to conservative learning rates, with substantially degraded performance observed under smaller update magnitudes, highlighting the limitations of purely accumulated gradient strategies when applied to deep residual architectures.

It is also observed that excessively large learning rates may lead to unstable training behavior for several optimizers, as reflected by the absence of valid results in certain configurations. This phenomenon further emphasizes the importance of maintaining a unified and bounded search space when comparing optimization strategies within DL models.

In short, the learning-rate sensitivity analysis confirms that different optimizers exhibit distinct convergence characteristics under the same DRN architecture. These results primarily serve as a unified hyperparameter selection reference rather than a direct comparison of forecasting performance. Based on the best-performing learning rate identified for each optimizer, subsequent experiments adopt the corresponding configuration to enable a fair evaluation of forecasting performance under different gradient-based optimization strategies.

Performance comparison of DRN under different gradient-based optimizers

To further examine the influence of optimization strategies on forecasting performance, the DRN model was evaluated under thirteen representative gradient-based optimizers using the corresponding learning-rate configurations determined from the sensitivity analysis. Table 4 summarizes the quantitative evaluation results on the ISO-NE dataset, while Fig. 6 provides a visual comparison of MAPE values to illustrate the relative performance differences among optimizers under identical training conditions.

As shown in Table 4; Fig. 6, noticeable variations in forecasting accuracy are observed even though the same DRN architecture and experimental settings are adopted. Classical stochastic gradient descent–based methods generally exhibit higher prediction errors compared with adaptive approaches. For instance, SGD achieves a MAPE of 0.020318, whereas the introduction of momentum improves performance to 0.018898, indicating that momentum-based updates enhance convergence stability. Among classical methods, NAG attains the strongest performance with a MAPE of 0.016330, suggesting that accelerated gradient mechanisms contribute to more effective optimization within the residual learning framework.

Optimizer algorithm	MAPE	MAE	MSE	RMSE	NMSE	R	R^2
SGD	0.020318	0.013239	0.000433	0.020806	0.028722	0.987106	0.971278
SGD-Momentum	0.018898	0.012208	0.000351	0.018733	0.023284	0.989899	0.976716
NAG	0.016330	0.010557	0.000296	0.017205	0.019640	0.990223	0.980360
AdaGrad	0.018937	0.012162	0.000359	0.018956	0.023844	0.988033	0.976156
AdaDelta	0.031948	0.020558	0.000898	0.029968	0.059589	0.970931	0.940411
RMSProp	0.018494	0.012008	0.000360	0.018985	0.023915	0.988223	0.976085
Adam	0.016278	0.010595	0.000309	0.017583	0.020514	0.989690	0.979486
Adamax	0.016630	0.010743	0.000294	0.017141	0.019495	0.990270	0.980505
Nadam	0.017690	0.011385	0.000331	0.018195	0.021968	0.989134	0.978032
AMSGrad	0.015756	0.010175	0.000278	0.016669	0.018437	0.990812	0.981563
AdamW	0.016602	0.010716	0.000303	0.017413	0.020120	0.990480	0.979880
RAdam	0.017109	0.011021	0.000303	0.017407	0.020105	0.989951	0.979895
AdaBelief	0.016336	0.010552	0.000284	0.016850	0.018838	0.990553	0.981162

Table 4. Performance comparison of DRN under different gradient-based optimizers on the ISO-NE dataset.

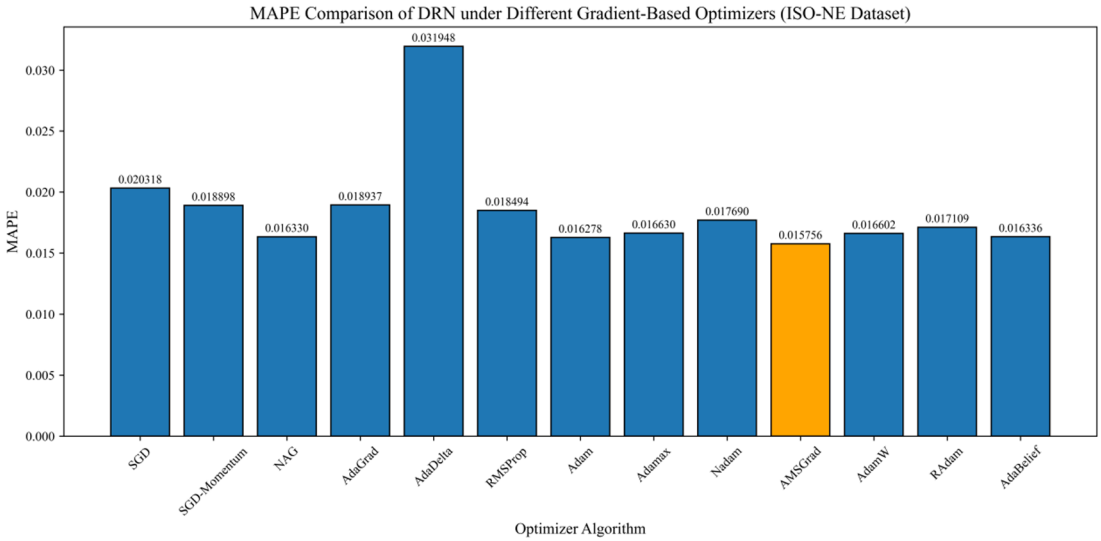

Fig. 6. MAPE comparison of DRN under different gradient-based optimizers on the ISO-NE dataset.

In contrast, adaptive gradient-based optimizers demonstrate more stable and consistently lower prediction errors. RMSProp and AdaGrad provide moderate improvements over classical approaches, while AdaDelta shows noticeably degraded performance, indicating that its accumulated update mechanism may be less suitable for deep residual architectures in this experimental setting. Within the Adam-family optimizers, Adam achieves the second-lowest MAPE value of 0.016278 following AMSGrad, indicating reliable convergence behavior under the DRN framework. Refined variants such as AdamW, Adamax, RAdam, and AdaBelief maintain competitive performance within a narrow accuracy range, suggesting that modifications to the adaptive update mechanism can preserve stable optimization dynamics without substantially altering convergence characteristics.

Among all evaluated methods, AMSGrad achieves the lowest MAPE value of 0.015756 together with the highest correlation metrics ($R = 0.990812$ and $R^2 = 0.981563$), indicating superior convergence stability under the DRN architecture. AdaBelief and Adamax also demonstrate strong performance, reflecting the effectiveness of refined adaptive strategies in improving optimization robustness. The clustering of most adaptive optimizers within a relatively narrow MAPE interval, as illustrated in Fig. 6, further suggests that optimizer selection influences convergence behavior even when structural factors remain unchanged.

Taken together, the comparative results indicate that optimizer choice has a measurable impact on DRN-based STLF performance. While classical gradient descent methods provide acceptable baseline accuracy, adaptive gradient-based optimizers—particularly AMSGrad—exhibit improved convergence stability and predictive performance under the unified experimental framework adopted in this study. Unlike many previous DRN-based STLF studies that adopt Adam as a default optimizer without systematic comparison, the present results demonstrate that AMSGrad can provide more stable convergence under the DRN framework.

Optimizer algorithm	MAPE	RMSE	MAE	MSE	NMSE	R	R^2
SGD	0.059559	0.049746	0.031838	0.002475	0.086920	0.959048	0.913080
SGD-momentum	0.054527	0.047531	0.027916	0.002259	0.079353	0.960828	0.920647
NAG	0.059803	0.049361	0.032160	0.002437	0.085581	0.961268	0.914419
AdaGrad	0.059418	0.050279	0.031233	0.002528	0.088792	0.957359	0.911208
AdaDelta	0.070161	0.058211	0.036986	0.003389	0.119019	0.941602	0.880981
RMSProp	0.057384	0.049013	0.029796	0.002402	0.084378	0.958517	0.915622
Adam	0.053519	0.047223	0.027487	0.002230	0.078326	0.960838	0.921674
Adamax	0.054368	0.045021	0.027147	0.002027	0.071191	0.964814	0.928809
Nadam	0.057242	0.051119	0.029541	0.002613	0.091783	0.953467	0.908217
AMSGrad	0.051152	0.046104	0.026392	0.002126	0.074657	0.962117	0.925343
AdamW	0.057474	0.050824	0.028926	0.002583	0.090730	0.954530	0.909270
RAdam	0.057834	0.049964	0.029953	0.002496	0.087682	0.957563	0.912318
AdaBelief	0.054198	0.045700	0.029288	0.002088	0.073356	0.965968	0.926644

Table 5. Performance comparison of DRN under different gradient-based optimizers on the MyPJ dataset.

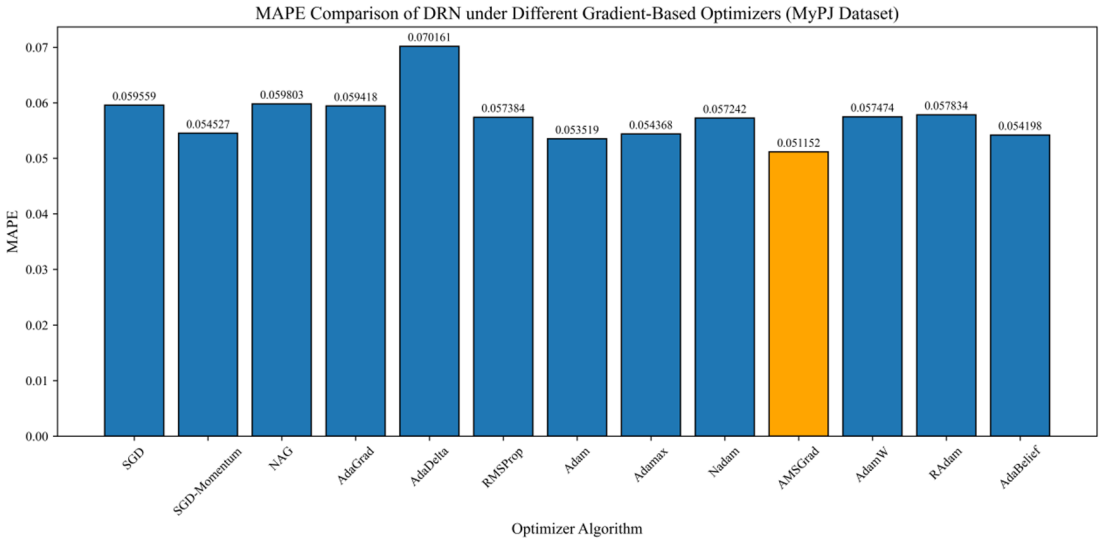

Fig. 7. MAPE comparison of DRN under different gradient-based optimizers on the MyPJ dataset.

Performance evaluation on the MyPJ dataset

DRN forecasting performance

To further examine the generalization capability of different optimization strategies under a tropical climate scenario, the DRN model was evaluated on the MyPJ dataset using the same experimental protocol adopted for the ISO-NE dataset. Table 5 summarizes the quantitative forecasting performance of thirteen gradient-based optimizers, while Fig. 7 provides a visual comparison of MAPE values to illustrate relative performance differences under identical architectural and training settings.

As shown in Table 5, noticeable variations in forecasting accuracy can be observed across different optimization strategies even though the DRN architecture remains unchanged. Classical stochastic gradient descent–based methods generally exhibit relatively higher prediction errors. For example, SGD achieves a MAPE of 0.059559, while the introduction of momentum improves performance to 0.054527, indicating that momentum-based updates help stabilize convergence. However, NAG does not demonstrate the same level of improvement on the MyPJ dataset, suggesting that accelerated gradient mechanisms may be more sensitive to the variability of tropical load patterns.

Among adaptive gradient-based optimizers, RMSProp, Adam, Adamax, AMSGrad, and AdaBelief demonstrate comparatively stronger forecasting performance. Adam achieves a MAPE of 0.053519 and consistently ranks among the top-performing optimizers, indicating reliable convergence behaviour across different climatic scenarios. Adamax and AdaBelief maintain competitive accuracy together with stronger correlation coefficients and lower absolute error metrics such as RMSE and MAE. Although their MAPE values are slightly higher than that of AMSGrad, the improved RMSE and correlation performance indicate that these optimizers capture the overall load trend more effectively. This divergence across evaluation metrics suggests that different adaptive

optimizers emphasize distinct error characteristics during training, where MAPE highlights relative percentage deviations while RMSE and MAE reflect absolute prediction stability.

In contrast, AdaGrad and AdaDelta show noticeably degraded performance, with higher MAPE and NMSE values, implying that optimizers relying heavily on accumulated historical gradients may struggle to adapt to the temporal variability present in the MyPJ dataset.

Notably, AMSGrad achieves the lowest forecasting error with a MAPE of 0.051152 and maintains strong correlation performance, indicating superior convergence stability under the DRN framework. When compared with the results obtained on the ISO-NE dataset, AMSGrad demonstrates consistent ranking and competitive accuracy across both climatic conditions. Adam also exhibits stable second-tier performance across datasets, reinforcing its role as a reliable baseline optimizer within the Adam-family. Although absolute forecasting errors differ due to dataset characteristics, the relative performance stability of AMSGrad and Adam suggests that adaptive moment-based optimization provides robust parameter updates that are less sensitive to data distribution differences.

Collectively, the results on the MyPJ dataset further confirm that adaptive gradient-based optimizers provide more stable and consistent forecasting performance than classical gradient descent methods. No single optimizer dominates across all evaluation metrics; instead, optimizers such as AMSGrad, Adamax, and AdaBelief exhibit complementary strengths in relative error control, absolute error reduction, and correlation stability. These findings highlight the importance of evaluating optimization strategies using multiple criteria rather than relying solely on a single performance indicator when developing reliable DRN–based forecasting models.

PCA-based meteorological feature analysis for PCA-DRN

PCA was applied to the meteorological variables in the MyPJ dataset to examine the underlying variance structure and reduce feature redundancy. Using a cumulative explained variance criterion of 90%, five principal components were retained, which together account for 94.12% of the total variance in the original meteorological feature set. Among them, PCA component 1 explains 37.88% of the variance, followed by PCA component 2 with 19.62%, PCA component 3 with 14.23%, PCA component 4 with 12.08%, and PCA component 5 with 8.54%. These results indicate that the retained components preserve the dominant variability of the original weather variables in a compact form.

The loadings of the retained principal components are summarized in Table 6, through which the relative contribution of each original weather variable can be analyzed. PCA component 1 is dominated by negative loadings from mean temperature (− 0.579888), minimum temperature (− 0.503639), and maximum temperature (− 0.442409), indicating that it primarily represents overall temperature-related variability. PCA component 2 is characterized by strong positive contributions from maximum wind speed (0.701837) and rainfall (0.442154), together with a moderate positive loading from maximum temperature (0.365551), suggesting an association with extreme wind conditions and precipitation-related patterns. PCA component 3 is mainly defined by a pronounced negative loading from maximum wind direction (− 0.946270), reflecting variability associated with wind direction. PCA component 4 exhibits a strong negative contribution from mean wind speed (− 0.792360), accompanied by positive contributions from maximum temperature (0.413782) and rainfall (0.381735), capturing interaction effects between wind intensity, temperature, and precipitation. For PCA component 5, rainfall (0.709699) and mean wind speed (0.387480) provide the dominant positive contributions, while a notable negative loading from maximum wind speed (− 0.512842) indicates rainfall–wind coupling behavior under varying wind strength conditions.

Overall, the extracted principal components provide a compact and informative representation of the meteorological feature space, preserving key variability patterns while reducing redundancy among correlated variables. This PCA-based feature structure offers a consistent and interpretable basis for the subsequent comparative evaluation of forecasting performance between DRN and PCA-DRN models presented in the following section.

PCA-DRN forecasting performance

To further evaluate the impact of meteorological feature compression on optimization behaviour, the PCA-DRN model was trained using the same set of gradient-based optimizers under identical experimental settings. Table 7 summarizes the quantitative forecasting results on the MyPJ dataset, while Fig. 8 presents a visual comparison of MAPE values across optimizers.

As shown in Table 7, the performance trend of PCA-DRN differs from that observed in the original DRN framework. While adaptive optimizers continue to outperform classical stochastic gradient descent–based

Weather variable	PCA component 1	PCA component 2	PCA component 3	PCA component 4	PCA component 5
Rainfall	0.326146	0.442154	0.099971	0.381735	0.709699
Mean temperature	− 0.579888	0.082323	− 0.091294	0.141974	0.086094
Minimum temperature	− 0.503639	− 0.203754	− 0.011570	0.147226	0.123478
Maximum temperature	− 0.442409	0.365551	− 0.130403	0.413782	− 0.051551
Mean wind speed	− 0.293040	0.318846	0.079191	− 0.792360	0.387480
Maximum wind speed	0.105115	0.701837	− 0.250668	− 0.087986	− 0.512842
Maximum wind direction	0.105159	− 0.168348	− 0.946270	− 0.075194	0.240546

Table 6. Loadings of meteorological variables on the retained principal components.

Optimizer algorithm	MAPE	MAE	MSE	RMSE	NMSE	R	R^2
SGD	0.056020	0.027607	0.002252	0.047453	0.079092	0.960414	0.920908
SGD-Momentum	0.052392	0.027191	0.002267	0.047609	0.079611	0.960122	0.920389
NAG	0.055327	0.029164	0.002314	0.048100	0.081264	0.959955	0.918736
AdaGrad	0.054421	0.027802	0.002250	0.047438	0.079043	0.960027	0.920957
AdaDelta	0.067889	0.034243	0.003032	0.055065	0.106501	0.945312	0.893499
RMSProp	0.053229	0.027012	0.002224	0.047162	0.078125	0.960153	0.921875
Adam	0.051109	0.027337	0.002228	0.047197	0.078240	0.960559	0.921760
Adamax	0.052843	0.027227	0.002048	0.045253	0.071927	0.963957	0.928073
Nadam	0.057073	0.030118	0.002744	0.052383	0.096381	0.951041	0.903619
AMSGrad	0.051813	0.027336	0.002145	0.046314	0.075340	0.962822	0.924660
AdamW	0.053616	0.028079	0.002196	0.046863	0.077138	0.961766	0.922862
RAdam	0.052401	0.028544	0.002272	0.047670	0.079816	0.961186	0.920184
AdaBelief	0.051037	0.026402	0.002070	0.045499	0.072711	0.963533	0.927289

Table 7. Performance comparison of PCA-DRN under different gradient-based optimizers on the MyPJ dataset.

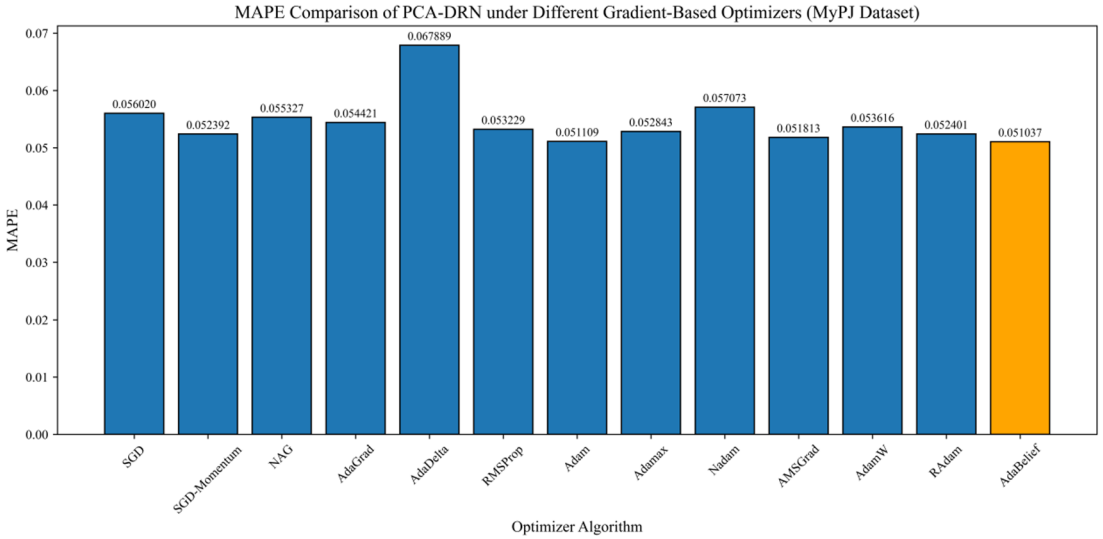

Fig. 8. MAPE comparison of PCA-DRN under different gradient-based optimizers on the MyPJ dataset.

methods, the relative ranking among Adam-family variants changes after PCA transformation. In the DRN configuration, AMSGrad achieved the lowest forecasting error; however, under PCA-DRN, AdaBelief attains the best performance with a MAPE of 0.051037, followed closely by Adam with a MAPE of 0.051109, whereas AMSGrad yields a slightly higher error of 0.051813. Although the performance gap among these adaptive optimizers remains relatively small, the shift in ranking suggests that optimizer behaviour may be influenced by the modified feature representation introduced by PCA.

One possible explanation is that PCA reduces redundancy and compresses correlated meteorological variables into a smaller set of orthogonal components, which alters the gradient distribution encountered during training. Under this transformed feature space, optimizers that adjust step sizes according to gradient variance estimation, such as AdaBelief, may better accommodate the smoother and decorrelated input structure. In contrast, the monotonic second-moment constraint imposed by AMSGrad, which previously enhanced convergence stability, may provide less relative advantage once feature redundancy has already been reduced through PCA. These observations should not be interpreted as a degradation of AMSGrad itself; rather, they indicate that optimizer effectiveness can vary depending on the input representation within the residual learning framework.

Compared with the DRN results, PCA-DRN demonstrates a more balanced performance distribution among adaptive optimizers, where AdaBelief, Adam, and Adamax achieve competitive accuracy within a narrow MAPE interval. This clustering effect suggests that PCA-based meteorological compression may stabilize learning dynamics across different adaptive update rules, thereby reducing sensitivity to optimizer selection. Meanwhile, classical optimizers such as SGD and AdaDelta continue to exhibit relatively higher prediction errors, reinforcing the importance of adaptive moment estimation strategies for deep residual forecasting models.

1st model	2nd model	MAPE ± SD (1st model)	MAPE ± SD (2nd model)	Mean difference	CI (95%)	*p*-value
DRN (Adam)	DRN (AMSGrad)	0.016278 ± 0.020375	0.015756 ± 0.019345	0.000522	[0.000264, 0.000782]	≈ 0

Table 8. Bootstrap statistical significance test results based on MAPE on the ISO-NE dataset.

1st model	2nd model	MAPE ± SD (1st model)	MAPE ± SD (2nd model)	Mean difference	CI (95%)	*p*-value
DRN (Adam)	PCA-DRN (Adam)	0.053519 ± 0.111717	0.051109 ± 0.093575	0.002410	[0.001061, 0.003854]	0.000300
DRN (AMSGrad)	PCA-DRN (AMSGrad)	0.051152 ± 0.099571	0.052525 ± 0.100998	− 0.001372	[− 0.002324, − 0.000440]	0.004400
DRN (AdaBelief)	DRN (AdaBelief)	0.054198 ± 0.090148	0.051037 ± 0.096386	0.003162	[0.002190, 0.004193]	≈ 0

Table 9. Bootstrap statistical significance test results based on MAPE on the MyPJ dataset.

Taken together, the PCA-DRN results suggest that feature representation can influence optimizer behaviour under the DRN framework, indicating that optimization strategy and input design should be considered jointly rather than independently when developing reliable STLF models. Previous STLF studies rarely examine how feature transformation affects optimizer behaviour. The present findings suggest that PCA-based meteorological compression can modify gradient characteristics, which may explain why AdaBelief becomes more effective under PCA-DRN compared with the original DRN configuration.

Statistical significance testing

Statistical significance testing is conducted to examine whether the observed differences in forecasting performance among models and optimization strategies are statistically meaningful rather than caused by random variation. The analysis is based on the MAPE, which serves as the primary evaluation metric throughout this study. A bootstrap resampling procedure with 10,000 iterations is employed to estimate paired performance differences, and the results are reported in terms of mean MAPE ± standard deviation (SD), mean difference, 95% CI, and corresponding p-values, as summarized in Tables 8 and 9.

For the ISO-NE dataset, the comparison between DRN optimized with Adam and AMSGrad shows a statistically significant improvement in favour of AMSGrad. The mean MAPE reduction of 0.000522, together with a confidence interval that does not cross zero, indicates that the performance gain reflects a consistent optimisation advantage rather than stochastic fluctuation. The extremely small p-value further confirms the robustness of this improvement.

On the MyPJ dataset, the bootstrap results reveal more diverse optimization behaviour under different feature representations. When comparing DRN and PCA-DRN under the Adam optimizer, PCA-DRN achieves a significantly lower MAPE, suggesting that PCA-based meteorological feature compression contributes to improved forecasting accuracy. In contrast, under AMSGrad, DRN slightly outperforms PCA-DRN, indicating that the stability advantages of AMSGrad within the original feature space may be reduced after PCA transformation. A more substantial improvement is observed for AdaBelief, where PCA-DRN demonstrates a clear and statistically significant reduction in MAPE compared with DRN, highlighting the ability of variance-adaptive optimizers to better exploit decorrelated feature representations.

Overall, the bootstrap analysis confirms that several performance differences observed across optimization strategies are statistically significant under the DRN framework. Rather than indicating a universally dominant optimizer, the results suggest that optimization effectiveness depends on both dataset characteristics and feature representation. These findings emphasize that optimizer selection and input design should be considered jointly when developing reliable DRN–based STLF models.

Summary

This section presents a comprehensive evaluation of architectural design, optimization strategies, and meteorological feature representation within the DRN-based STLF framework across both temperate (ISO-NE) and tropical (MyPJ) climatic conditions.

The baseline comparison conducted on the temperate ISO-NE dataset demonstrates that residual learning provides clear structural advantages over conventional DL architectures. Under identical training conditions using the Adam optimizer, the DRN framework achieves the lowest forecasting error among all evaluated models, indicating that hierarchical residual feature extraction contributes to improved convergence stability and predictive accuracy. These results establish a reliable architectural reference prior to the investigation of optimizer behaviour across different climatic scenarios.

Subsequent learning-rate sensitivity analysis reveals that different gradient-based optimizers exhibit distinct convergence characteristics even under a unified DRN architecture. Classical stochastic gradient descent–based methods show stronger dependence on learning-rate selection, whereas adaptive optimizers maintain relatively stable performance across a broader range of learning rates. The sensitivity analysis therefore provides a consistent hyperparameter selection basis for fair optimizer comparison rather than serving as a direct performance evaluation.

Performance comparisons on both the temperate ISO-NE dataset and the tropical MyPJ dataset further demonstrate that optimizer choice has a measurable influence on forecasting accuracy. Adaptive moment-based optimizers consistently outperform classical gradient descent approaches across climatic conditions, with

AMSGrad achieving the lowest MAPE under the original DRN configuration. However, the results also indicate that no single optimizer universally dominates across all scenarios. While AMSGrad exhibits strong convergence stability within the original feature space, AdaBelief becomes more effective after PCA-based meteorological feature compression on the tropical MyPJ dataset, suggesting that optimizer effectiveness depends on the interaction between optimization dynamics, climatic characteristics, and input representation.

The PCA-based meteorological feature analysis performed on the tropical MyPJ dataset confirms that a compact set of principal components can preserve dominant weather variability while reducing redundancy among correlated variables. This transformation alters the gradient characteristics encountered during training, leading to observable changes in optimizer ranking under the PCA-DRN framework. The findings highlight that feature engineering and optimization strategies should be considered jointly rather than independently when developing DRN-based forecasting models for datasets with diverse climatic conditions.

Finally, bootstrap statistical significance testing verifies that several observed performance improvements are statistically meaningful. On the temperate ISO-NE dataset, AMSGrad provides a statistically significant improvement over Adam, indicating more stable optimisation behaviour under residual learning. On the tropical MyPJ dataset, PCA-based feature compression leads to significant performance gains under certain adaptive optimizers, particularly AdaBelief, demonstrating that optimisation effectiveness is influenced by both climatic variability and feature representation.

In summary, the experimental results demonstrate that the forecasting performance of DRN-based models is governed by a combination of architectural design, optimizer behaviour, climatic characteristics, and meteorological feature representation. Across both temperate and tropical datasets, adaptive gradient-based optimizers provide more stable convergence behaviour than classical methods, while the effectiveness of individual optimizers varies depending on the input representation. These findings provide empirical evidence that systematic optimizer evaluation, together with appropriate feature transformation under different climatic conditions, plays a critical role in enhancing the reliability and stability of DRN–based STLF.

Conclusion

This study presents a systematic comparative evaluation of gradient-based optimization algorithms for DRN–based STLF under different climatic conditions and feature-representation settings. Unlike most existing DRN-based STLF studies that adopt Adam as a default training strategy, this work investigates thirteen representative optimization algorithms within a unified experimental framework, covering both the original DRN and its PCA-enhanced variant. Experimental validation is conducted using real-world electricity load datasets from the temperate ISO-NE system and the tropical MyPJ region, enabling a comprehensive analysis across distinct climatic characteristics.

The results demonstrate that residual learning provides a robust architectural foundation for accurate load forecasting, achieving superior performance compared with several widely adopted DL baselines under consistent training settings. Through learning-rate sensitivity analysis and controlled optimizer evaluation, adaptive gradient-based optimizers generally provide more stable convergence behaviour than classical stochastic gradient descent–based methods.

Across both temperate and tropical datasets, AMSGrad achieves the most consistent and lowest forecasting error within the original DRN framework, indicating strong cross-climate optimization stability under deep residual learning. This finding highlights that AMSGrad offers reliable convergence behaviour when the original meteorological feature space is preserved. However, the results also reveal that optimizer effectiveness is not universal and depends strongly on feature representation. After PCA-based meteorological feature compression, the relative advantage of AMSGrad becomes less pronounced, while AdaBelief demonstrates improved forecasting performance under the PCA-DRN configuration. This contrast suggests that PCA transformation reshapes the gradient landscape by reducing feature redundancy and altering gradient variance statistics, thereby changing the effectiveness of adaptive optimization strategies rather than indicating a degradation of AMSGrad itself. These observations emphasize that optimization strategy and feature design should be considered jointly rather than independently when developing DRN-based forecasting models.

Bootstrap-based statistical significance testing further confirms that several observed performance improvements are statistically meaningful. In particular, AMSGrad exhibits a significant advantage over Adam on the temperate ISO-NE dataset, while PCA-based feature compression leads to significant forecasting improvements under specific adaptive optimizers on the tropical MyPJ dataset. These results provide empirical evidence that systematic optimizer selection can enhance both training stability and forecasting accuracy in residual learning frameworks.

Although the proposed evaluation framework provides valuable insights into optimizer behaviour for DRN-based STLF, several directions remain open for future research. First, the present work focuses primarily on accuracy-oriented evaluation under a unified experimental protocol, where runtime efficiency and convergence speed (time-to-target error) were not explicitly investigated in order to maintain a controlled comparison of forecasting performance. Future studies may therefore extend this work by incorporating computational-efficiency analysis, including runtime benchmarking and convergence dynamics, to provide a more comprehensive understanding of optimizer behaviour in DRN-based forecasting models. Second, the present work considers static optimization settings under fixed data distributions. In real-world smart-grid environments, electricity demand patterns may evolve over time due to changing generation modalities, demand-side behaviour, and environmental factors. Future studies may therefore investigate adaptive learning strategies capable of handling concept drift and distributional shifts, such as transfer-learning–enabled forecasting frameworks or adaptive ensemble approaches[48]. Third, the integration of emerging intelligent infrastructure—such as digital twin systems and 5G-enabled industrial monitoring environments—may provide new opportunities for real-time model adaptation and immersive visualization of forecasting dynamics[49]. Exploring these directions could

further enhance the practical deployment and scalability of DRN-based forecasting models in next-generation smart energy systems.

In conclusion, this study demonstrates that optimizer selection plays a critical yet previously underexplored role in DRN-based STLF. By jointly examining architectural design, optimization strategies, climatic variability, and meteorological feature representation, the findings contribute to a more comprehensive understanding of how training dynamics influence forecasting performance. The proposed unified evaluation framework provides a practical reference for selecting appropriate optimization strategies in deep residual forecasting models and offers a foundation for future research toward adaptive, data-driven smart-grid forecasting systems.

Data availability

The datasets generated and/or analysed during the current study are not publicly available due to licensing and institutional restrictions, but are available from the corresponding author upon reasonable request.

Received: 16 January 2026; Accepted: 23 March 2026
Published online: 25 March 2026

References

1. Ahmad, F. A., Liu, J., Hashim, F. & Samsudin, K. Short-term load forecasting utilizing a combination model: a brief review. *Int. J. Technol.* **15**, 121–129. https://doi.org/10.14716/ijtech.v15i1.5543 (2024).
2. Hobbs, B. F. et al. Analysis of the value for unit commitment of improved load forecasts. *IEEE Trans. Power Syst.* **14**, 1342–1348. https://doi.org/10.1109/59.801894 (1999).
3. Wang, Y. & Wu, L. Improving economic values of day-ahead load forecasts to real-time power system operations. *IET Gener Transm Distrib.* **11**, 4238–4247. https://doi.org/10.1049/iet-gtd.2017.0517 (2017).
4. Akhtar, S. et al. Short-term load forecasting models: A review of challenges, progress, and the road ahead. *Energies* **16**, 4060. https://doi.org/10.3390/en16104060 (2023).
5. Hippert, H. S., Pedreira, C. E. & Souza, R. C. Neural networks for short-term load forecasting: A review and evaluation. *IEEE Trans. Power Syst.* **16**, 44–55. https://doi.org/10.1109/59.910780 (2001).
6. Ceperic, E., Ceperic, V. & Baric, A. A strategy for short-term load forecasting by support vector regression machines. *IEEE Trans. Power Syst.* **28**, 4356–4364. https://doi.org/10.1109/TPWRS.2013.2269803 (2013).
7. Kuster, C., Rezgui, Y. & Mourshed, M. Electrical load forecasting models: A critical systematic review. *Sustain. Cities Soc.* **35**, 257–270. https://doi.org/10.1016/j.scs.2017.08.009 (2017).
8. Cecati, C., Kolbusz, J., Różycki, P., Siano, P. & Wilamowski, B. M. A novel RBF training algorithm for short-term electric load forecasting and comparative studies. *IEEE Trans. Ind. Electron.* **62**, 6519–6529. https://doi.org/10.1109/TIE.2015.2424399 (2015).
9. Chen, Y. et al. Short-term load forecasting: Similar day-based wavelet neural networks. *IEEE Trans. Power Syst.* **25**, 322–330. https://doi.org/10.1109/TPWRS.2009.2030426 (2009).
10. Zhao, Y., Luh, P. B., Bomgardner, C. & Beerel, G. H. Short-term load forecasting: Multi-level wavelet neural networks with holiday corrections. In *Proc. IEEE Power Energy Soc. Gen. Meeting*, 1–7. https://doi.org/10.1109/PES.2009.5275304 (2009).
11. Eren, Y. & Küçükdemiral, İ. A comprehensive review on deep learning approaches for short-term load forecasting. *Renew. Sustain. Energy Rev.* **189**, 114031. https://doi.org/10.1016/j.rser.2023.114031 (2024).
12. Li, L., Ota, K. & Dong, M. Everything is image: CNN-based short-term electrical load forecasting for smart grid. In *Proc. 14th Int. Symp. Pervasive Syst. Algorithms Netw. (ISPAN-FCST-ISCC)*, 344–351. https://doi.org/10.1109/ISPAN-FCST-ISCC.2017.78 (2017).
13. Jurado, M., Samper, M. & Rosés, R. An improved encoder-decoder-based CNN model for probabilistic short-term load and PV forecasting. *Electr. Power Syst. Res.* **217**, 109153. https://doi.org/10.1016/j.epsr.2023.109153 (2023).
14. Narayan, A. & Hipel, K. W. Long short-term memory networks for short-term electric load forecasting. In *Proc. IEEE Int. Conf. Syst. Man Cybern.*, 2573–2578. https://doi.org/10.1109/SMC.2017.8123012 (2017).
15. Bento, P., Pombo, J., Mariano, S. & Calado, M. R. Short-term load forecasting using optimized LSTM networks via improved bat algorithm. In *Proc. Int. Conf. Intell. Syst.*, 351–357. https://doi.org/10.1109/IS.2018.8710498 (2018).
16. Ran, P., Dong, K., Liu, X. & Wang, J. Short-term load forecasting based on CEEMDAN and Transformer. *Electr. Power Syst. Res.* **214**, 108885. https://doi.org/10.1016/j.epsr.2022.108885 (2023).
17. Jiang, B., Liu, Y., Geng, H., Zeng, H. & Ding, J. A transformer-based method with wide attention range for enhanced short-term load forecasting. In *Proc. 4th Int. Conf. Smart Power Internet Energy Syst.*, 1684–1690 (2022).
18. Li, S., Zhang, W. & Wang, P. TS2ARCformer: A multi-dimensional time series forecasting framework for short-term load prediction. *Energies* **16**, 5825. https://doi.org/10.3390/en16155825 (2023).
19. Chen, Y., Wang, Y., Liu, X. & Huang, J. Short-term load forecasting for industrial users based on Transformer–LSTM hybrid model. In *Proc. IEEE 5th Int. Electr. Energy Conf. (CIEEC)*, 2470–2475. https://doi.org/10.1109/CIEEC54735.2022.9846658 (2022).
20. Guo, W., Liu, S., Weng, L. & Liang, X. Power grid load forecasting using a CNN–LSTM network based on a multi-modal attention mechanism. *Appl. Sci.* **15**, 2435. https://doi.org/10.3390/app15052435 (2025).
21. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 770–778. https://doi.org/10.48550/arXiv.1512.03385 (2016).
22. Chen, K. et al. Short-term load forecasting with deep residual networks. *IEEE Trans. Smart Grid*. **10**, 3943–3952. https://doi.org/10.1109/TSG.2018.2844307 (2018).
23. Kondaiah, V. Y. & Saravanan, B. Short-term load forecasting with deep learning. In *Proc. Innov. Power Adv. Comput. Technol. (i-PACT)*, 1–5. https://doi.org/10.1109/i-PACT52855.2021.9696634 (2021).
24. Xu, Q., Yang, X. & Huang, X. Ensemble residual networks for short-term load forecasting. *IEEE Access.* **8**, 64750–64759. https://doi.org/10.1109/ACCESS.2020.2984722 (2020).
25. Chen, W., Han, G., Zhu, H., Liao, L. & Zhao, W. Deep ResNet-based ensemble model for short-term load forecasting in protection system of smart grid. *Sustainability* **14**, 16894. https://doi.org/10.3390/su142416894 (2022).
26. Kondaiah, V. Y. & Saravanan, B. A modified deep residual network for short-term load forecasting. *Front. Energy Res.* **10**, 1038819. https://doi.org/10.3389/fenrg.2022.1038819 (2022).
27. Ding, A., Liu, T. & Zou, X. Integration of ensemble GoogLeNet and modified deep residual networks for short-term load forecasting. *Electronics* **10**, 2455. https://doi.org/10.3390/electronics10202455 (2021).
28. Sheng, Z., Wang, H., Chen, G., Zhou, B. & Sun, J. Convolutional residual network for short-term load forecasting. *Appl. Intell.* **51**, 2485–2499. https://doi.org/10.1007/s10489-020-01932-9 (2021).
29. Tian, Y., Yu, S., Wen, M., Zhang, K. & Chen, Y. Short-term load forecasting scheme based on improved deep residual network and LSTM. In *Proc. CIRED Berlin Workshop*, 117–120. https://doi.org/10.1049/oap-cired.2021.0257 (2020).
30. Li, H., Zhang, P. & Li, C. Short-term load forecasting for distribution substations based on residual neural networks and long short-term memory neural networks with attention mechanism. *J. Phys. Conf. Ser.* **2030**, 012087. https://doi.org/10.1088/1742-6596/2030/1/012087 (2021).

31. Sheng, Z., An, Z., Wang, H., Chen, G. & Tian, K. Residual LSTM-based short-term load forecasting. *Appl. Soft Comput.* **144**, 110461. https://doi.org/10.1016/j.asoc.2023.110461 (2023).
32. Liu, J., Ahmad, F. A., Samsudin, K. & Hashim, F. Ab Kadir, M. Z. A. Optimizing deep residual networks for short-term load forecasting with multidimensional weather data and principal component analysis. *IEEE Access.* **13**, 158614–158631. https://doi.org/10.1109/ACCESS.2025.3607330 (2025).
33. Makinde, A. Optimizing time series forecasting: A comparative study of Adam and Nesterov accelerated gradient on LSTM and GRU networks using stock market data. *arXiv* **2410.01843** https://doi.org/10.48550/arXiv.2410.01843 (2024).
34. Dai, L. Performance analysis of deep learning-based electric load forecasting model with particle swarm optimization. *Heliyon* **10**, e35273. https://doi.org/10.1016/j.heliyon.2024.e35273 (2024).
35. Zhou, K., Wang, W., Hu, T. & Deng, K. Time series forecasting and classification models based on recurrent networks with attention mechanism and generative adversarial networks. *Sensors* **20**, 7211. https://doi.org/10.3390/s20247211 (2020).
36. Dogo, E. M., Afolabi, O. J., Nwulu, N. I., Twala, B. & Aigbavboa, C. O. A comparative analysis of gradient descent-based optimization algorithms on convolutional neural networks. In *Proc. Int. Conf. Comput. Tech. Electron. Mech. Syst. (CTEMS)*, 92–99. https://doi.org/10.1109/CTEMS.2018.8769211 (2018).
37. Mustapha, A., Mohamed, L. & Ali, K. Comparative study of optimization techniques in deep learning: Application in the ophthalmology field. *J. Phys. Conf. Ser.* **1743**, 012002. https://doi.org/10.1088/1742-6596/1743/1/012002 (2021).
38. Murthy, A. et al. Optimizing convolutional neural networks: A comparative study of gradient-descent, Adam, and RMSprop optimizers for accuracy and loss in apple leaf disease detection. In *Proc. 2nd Int. Conf. Networks Multimedia Inf. Technol. (NMITCON)*, 1–6. https://doi.org/10.1109/NMITCON62075.2024.10699138 (2024).
39. Sembiring, Z., Rani, K. N. A. & Amir, A. A comparative study of gradient descent methods in deep learning using body motion dataset. *Pertanika J. Sci. Technol.* **33** https://doi.org/10.47836/pjst.33.4.13 (2025).
40. Sashank, J. R., Satyen, K. & Kumar, S. On the convergence of Adam and beyond. In *Proc. Int. Conf. Learn. Representations (ICLR)*. https://doi.org/10.48550/arXiv.1904.09237 (2018).
41. Loshchilov, I. & Hutter, F. Fixing weight decay regularization in Adam. arXiv 1711.05101. https://doi.org/10.48550/arXiv.1711.05101 (2017).
42. Liu, L. et al. On the variance of the adaptive learning rate and beyond. arXiv:1908.03265. https://doi.org/10.48550/arXiv.1908.03265 (2019).
43. Zhuang, J. et al. AdaBelief optimizer: Adapting stepsizes by the belief in observed gradients. *Adv. Neural Inf. Process. Syst.* **33**, 18795–18806 (2020).
44. Shi, H., Xu, M. & Li, R. Deep learning for household load forecasting—A novel pooling deep RNN. *IEEE Trans. Smart Grid.* **9**, 5271–5280. https://doi.org/10.1109/TSG.2017.2686012 (2017).
45. Huang, G. et al. Snapshot ensembles: Train 1, get m for free. arXiv 1704.00109. https://doi.org/10.48550/arXiv.1704.00109 (2017).
46. Khoshkangini, R., Tajgardan, M., Lundström, J., Rabbani, M. & Tegnered, D. A snapshot-stacked ensemble and optimization approach for vehicle breakdown prediction. *Sensors* **23**, 5621. https://doi.org/10.3390/s23125621 (2023).
47. Johnston, M. G. & Faulkner, C. A bootstrap approach is a superior statistical method for the comparison of non-normal data with differing variances. *New. Phytol.* **230**, 23–26 (2021).
48. Azeem, A., Ismail, I., Jameel, S. M. & Danyaro, K. U. Transfer-learning enabled adaptive framework for load forecasting under concept-drift challenges in smart-grids across different-generation-modalities. *Energy Rep.* **12**, 3519–3532. https://doi.org/10.1016/j.egyr.2024.09.040 (2024).
49. Azeem, A. et al. A 5G-enabled digital twin framework for AI-powered smart industry monitoring and immersive visualization. In *2025 IEEE 17th Malaysia International Conference on Communication (MICC)*, 120–124, IEEE. https://doi.org/10.1109/MICC66164.2025.11211484 (2025).

Author contributions

J.L. and F.A.A. contributed to the conceptualization of the study. J.L. developed the methodology and conducted the investigation with F.A.A. J.L. prepared the original draft. J.L., F.A.A., K.S., F.H., and M.Z.A.A.K. contributed to the review and editing of the manuscript. F.A.A., K.S., F.H., and M.Z.A.A.K. provided supervision. All authors have read and agreed to the published version of the manuscript.

Funding

The authors declare that no external funding was received for this study.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-026-45829-y.

Correspondence and requests for materials should be addressed to F.A.A.