scientific reports

Check for updates

OPEN

Mini-batch size sensitivity in deep residual networks for short-term load forecasting: an empirical study

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

Mini-batch size is a fundamental yet often empirically selected training configuration in deep learning (DL)–based short-term load forecasting (STLF). Despite the widespread adoption of deep residual networks (DRNs) for STLF, the sensitivity of forecasting performance to mini-batch size has received limited systematic investigation. This study presents a comprehensive empirical analysis of mini-batch size effects in DRN-based STLF, using the original DRN and the Principal Component Analysis–Deep Residual Network (PCA-DRN) as representative frameworks. Experiments are conducted on real-world electricity load datasets representing different climatic conditions under a unified experimental setting, where network architecture, loss function, optimizer, and training epochs are strictly controlled and only the mini-batch size is varied. The results demonstrate that both DRN and PCA-DRN exhibit pronounced sensitivity to mini-batch size, with medium-scale batches achieving superior forecasting performance. In particular, a mini-batch size of 64 consistently yields the best results, while a batch size of 32 provides a robust alternative. Comparative evaluations with representative DL baseline models further highlight the effectiveness of residual-based forecasting. Bootstrap-based statistical significance testing confirms that the observed performance improvements are reliable and reproducible. Overall, this study provides practical guidance on mini-batch size selection for residual-network-based STLF and offers new insights into training configuration effects in DL-driven STLF across different climatic electricity systems.

Keywords Bootstrap statistical analysis, Deep residual networks, Mini-batch size, PCA-based feature compression, Short-term load forecasting

Abbreviations

Abbreviation	Full name
AI	Artificial Intelligence
ANN	Artificial neural network
CI	Confidence interval
CNN	Convolutional neural network
CNN-LSTM-MMA	CNN–LSTM with multi-modal attention
Conv1D	One-dimensional convolution
CRN	Convolutional residual network
DL	Deep learning
DNN	Deep neural network
DRN	Deep residual network
ERN	Ensemble residual network
FC	Fully connected
ISO-NE	New England Independent System Operator
LSTM	Long short-term memory
MAE	Mean absolute error
MAPE	Mean absolute percentage error
MSE	Mean squared error

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

MW	Megawatt
MyPJ	Malaysia Petaling Jaya
NMSE	Normalized mean squared error
PCA	Principal component analysis
PCA-DRN	Principal component analysis–deep residual network
R	Correlation coefficient
R^2	Coefficient of determination
ReLU	Rectified linear unit
ResBlock	Residual block
ResNet	Residual network
RMSE	Root mean squared error
RNN	Recurrent neural network
SELU	Scaled exponential linear unit
STLF	Short-term load forecasting
TCN	Temporal convolutional network

The reliability and operational efficiency of modern electricity systems depend on accurate Short-Term Load Forecasting (STLF). By utilizing STLF's precise demand forecasts over time horizons ranging from the next 24 h to a week, utilities may enhance generation scheduling, unit commitment, and energy trading tactics. By enabling improved grid design, operation, and management, accurate load forecasts improve supply stability, reduce operating costs, and boost overall energy efficiency. Due to the continuous increase in energy demand and the expanding variety of consumption patterns, STLF has become much more complicated and significant[1,2]. For big utilities, for instance, a 1% increase in forecast accuracy can save millions of dollars annually through improved resource allocation and reduced reserve needs[3,4]. Improvements in predicting accuracy, no matter how little, may significantly boost the economy.

Currently, there are two main approaches to STLF problems: traditional techniques and artificial intelligence (AI)-based solutions. Conventional approaches frequently make use of statistical models and traditional machine learning techniques[5], which laid the foundation for early demand modeling research and provided useful data on load behavior. However, these methods have a limited ability to capture highly variable and nonlinear demand patterns, sometimes rely on simplistic assumptions about load dynamics, and grow more prone to overfitting as input dimensionality rises[6,7].

To overcome these limitations, STLF research has made substantial use of AI-based techniques, particularly artificial neural networks (ANNs). Through deep learning (DL), ANNs may improve forecasting accuracy, eliminate overfitting to some extent, and approximate complex load dynamics[8]. Overfitting may still be a problem, though, if more input variables, hidden neurons, or network layers are introduced. To solve this issue, several ANN modifications have been investigated in the framework of STLF[9–11]. Despite these efforts, standard ANNs and their extensions, which are by nature shallow structures, show insufficient capacity to represent extremely complex load patterns. Because they allow for more expressive modeling of complex load dynamics and hierarchical feature learning, deep neural networks (DNNs) with many hidden layers have been more prominent in STLF research. DL based forecasting models have also been widely applied in domains beyond power systems. For example, neural network architectures have demonstrated strong predictive capability in financial time-series forecasting, where complex nonlinear market dynamics and temporal dependencies must be captured for stock price prediction and trend analysis[12,13]. This development has led to a paradigm change from basic ANN-based models to complex DL frameworks that can capture complex temporal and spatial correlations from a range of data sources[14].

In recent years, classic shallow designs in STLF have been substantially superseded by neural models with deeper structures and a variety of input combinations. Convolutional Neural Networks (CNNs) use localized convolutional operations to efficiently extract short-term temporal load patterns; nevertheless, their capacity to represent long-range temporal dependencies is still restricted, and training becomes more challenging as network depth increases[15,16]. Temporal Convolutional Networks (TCNs) are an extension of traditional CNNs that use dilated and causal convolutions to model longer-range temporal dependencies in parallel and expand the receptive field. However, their performance can rely on careful architectural design, and capturing very long-term dependencies may still require deeper or more complex network configurations[17,18]. Although memory cells and gating mechanisms are used by recurrent neural networks (RNNs), especially Long Short-Term Memory (LSTM) networks, to explicitly capture temporal dependencies and mitigate the vanishing gradient problem, their intrinsically sequential computations decrease efficiency when processing lengthy input sequences[19,20]. Transformer-based architectures that use self-attention techniques have attracted a lot of interest lately since they can fully parallelize the modeling of long-range relationships[21–23]. Despite these benefits, large computing costs and uneven training behavior in deep configurations continue to be major obstacles.

To further increase forecasting accuracy and generalization, hybrid architectures combining convolutional, recurrent, and attention-based techniques have been proposed. Two prominent examples that aim to use complementary skills across several modeling paradigms are Transformer–LSTM[24] and CNN-LSTM with multi-modal attention (CNN-LSTM-MMA)[25]. Although these hybrid approaches have demonstrated improved predictive performance, they still struggle with training stability and computing efficiency. These challenges underscore the need for a more dependable, efficient, and scalable DL framework for STLF.

Residual learning with identity shortcut connections was originally introduced in the residual network (ResNet)[26] architecture to mitigate the vanishing gradient problem and stabilize the training of DNNs. Building on this concept, Chen et al.[27] proposed the Deep Residual Network (DRN) for STLF, demonstrating that an appropriate balance between network depth, optimization stability, and representation capacity can lead to

reliable high-performance forecasting. Following the original DRN formulation, a wide range of DRN-based variants have been developed to further enhance predictive accuracy and robustness, including task-specific residual architectures designed for STLF[28], ensemble-augmented DRN models employing snapshot or residual ensemble strategies[29,30], data-driven DRN frameworks with adaptive feature selection mechanisms[31], structurally refined ResNets incorporating multi-scale or inception-inspired modules[32,33], and hybrid DRN architectures that integrate recurrent or attention-based components to strengthen temporal dependency modeling[34–36]. In addition, the Principal Component Analysis–Deep Residual Network (PCA-DRN) has been introduced to incorporate multiple meteorological variables through dimensionality reduction techniques[37].

Despite the growing body of research on DRN-based STLF, most existing studies primarily focus on architectural refinement, feature-level enhancement, or ensemble learning strategies to improve predictive accuracy. In contrast, the role of training configuration parameters, particularly mini-batch size, has received limited systematic investigation within the context of DRNs for STLF. In practical applications, mini-batch size is often selected empirically and treated as a fixed hyperparameter, despite its potential influence on training performance and forecasting accuracy. Moreover, it remains unclear whether similar batch-size sensitivity characteristics persist when ResNets are extended to incorporate multiple meteorological variables, as in the PCA-DRN framework. These gaps motivate a comprehensive empirical examination of mini-batch size sensitivity in both DRN and PCA-DRN models under a unified experimental setting.

The main contributions of this study can be summarized as follows. First, this work presents a systematic empirical investigation of mini-batch size sensitivity in DRN-based STLF, explicitly treating mini-batch size as a core training factor rather than a fixed hyperparameter. Second, the analysis is conducted consistently on both the original DRN and the PCA-DRN frameworks using real-world datasets representing both temperate and tropical climatic conditions. This design enables a direct comparison of batch size effects under different meteorological input representations while maintaining identical residual learning structures. Third, comprehensive experimental evaluation, including comparative analysis with representative DL baseline models and bootstrap-based statistical significance testing, is performed to ensure the robustness and reproducibility of the observed performance differences. Finally, this study provides practical guidance for selecting appropriate mini-batch sizes when training residual-based forecasting models for STLF applications.

The remainder of this paper is organized as follows. Section 2 reviews DL-based STLF methods, with an emphasis on convolutional, recurrent, attention-based, and residual learning frameworks. Section 3 describes the research methodology, including dataset description, model architectures, experimental design, and evaluation metrics. Section 4 presents the experimental results and discussion, analyzing the effects of different mini-batch size configurations on the forecasting performance of DRN and PCA-DRN, together with comparative evaluation and statistical significance assessment. Finally, Sect. 5 concludes the paper by summarizing the main findings, discussing limitations, and outlining directions for future research.

Deep learning-powered short-term load forecasting methods

Because of the rapid advancement of sensor technologies, advanced metering infrastructures, and modern high-performance computing platforms, DL techniques have become more prevalent in STLF. Because DNNs are more capable of approximating nonlinear interactions and learning complex temporal dependencies than classic statistical and shallow learning approaches, they are often employed in contemporary STLF research. Modern DL-based forecasting methods are frequently categorized using convolution-based, recurrent-based, attention-driven, and hybrid modeling frameworks, each of which highlights certain facets of load time-series data.

CNNs have been thoroughly studied in STLF applications due to their excellent capacity to capture local temporal patterns and short-range interdependence. Li et al.[15] made it possible for convolutional kernels to take advantage of spatial correlations by converting load sequences into image-like representations. This led to significant gains throughout a variety of predicting horizons. However, the additional preprocessing processes and dual-branch design significantly increased structural complexity, which limited real-time deployment. Jurado et al.[16] created an encoder–decoder CNN system with Monte Carlo Dropout and probabilistic density estimation to handle uncertainty quantification. Reduced accuracy during high load times demonstrated inadequate resilience under sudden demand fluctuations, despite the fact that improved performance over traditional recurrent models was attained. By adding causal and dilated convolutions to capture long-range temporal connections, TCNs have been proposed as an extension of traditional convolutional models for time-series forecasting. In order to enhance forecasting accuracy, Tang et al.[17] integrated channel and temporal attention mechanisms into a TCN framework to describe cumulative weather impacts and diverse feature significance. However, the additional attention modules increased architectural complexity. Liu et al.[18] created a temporal depthwise convolutional network based on TCN to improve computational efficiency. This network preserved temporal modeling capability while reducing parameter redundancy through depthwise separable convolution. However, the model is still primarily convolution-driven and may be less adaptable when representing extremely complex temporal dynamics.

Sequence modeling remains a central challenge in STLF, particularly under complex seasonal dynamics. Models based on recurrent architectures, including LSTM variants, have demonstrated strong predictive capability when sufficient historical information is available[19]. Nevertheless, forecasting accuracy may decline in practice when key exogenous variables, such as weather-related factors, are not explicitly considered. Bento et al.[20] attempted to alleviate this limitation through metaheuristic-driven hyperparameter optimization, although the resulting increase in computational demand raises concerns regarding scalability.

The ability of attention mechanisms and Transformer-based designs to leverage global self-attention to record long-range temporal correlations has recently attracted a lot of interest, in contrast to recurrent structures. Ran et al.[21] used transformer networks with empirical mode decomposition to enhance temporal feature extraction; however, their reliance on predefined decomposition parameters and lengthy training time hampered its ability

to adapt to unexpected datasets. Jiang et al.[22] expanded the attention receptive region to increase prediction accuracy, however this resulted in a significant memory cost. Li et al.[23] further this field of study by introducing TS2ARCformer, which combines contextual encoding, cross-dimensional attention, and autoregressive components. Despite performance gains on test datasets, the hierarchical attention structure and autoregressive modeling greatly increased architectural complexity, making practical STLF implementation challenging.

In an effort to mitigate the inherent shortcomings of single-paradigm systems, recent research has focused more on hybrid DL frameworks that integrate complementary modeling techniques. Chen et al.[24] introduced a Transformer–LSTM hybrid architecture for industrial STLF that employed self-attention to capture global temporal interactions and LSTM layers to mimic sequential dependencies. Despite continuous performance gains, the cascaded structure necessitated meticulous hyperparameter adjustment and increased computing cost. Similar to this, Guo et al.[25] developed an enhanced CNN–LSTM framework with multi-modal attention to adaptively fuse a variety of inputs, including load demand and meteorological data. Even though prediction accuracy was improved, the attention-based fusion approach increased computational complexity and imposed stricter requirements on data availability and quality.

Deeper and more complex network designs can exacerbate training instability, gradient degradation, and performance saturation, even when convolution-based, recurrent-based, attention-based, and hybrid DL models achieve notable performance gains. These challenges underscore the need for learning paradigms that can support deep structures while maintaining consistent optimization behavior. DRNs have emerged as an effective solution in this scenario by incorporating residual connections that facilitate gradient propagation and enable scalable depth extension.

In STLF applications, DRN-based models have demonstrated notable advantages in characterizing intricate and highly nonlinear load patterns. Early research focused mostly on feasibility evaluation and comparative analysis with conventional DL techniques. Chen et al.[27] coupled residual learning with ensemble approaches to increase robustness and generalization, although at a greater computational cost. Kondaiah et al.[28] introduced a task-oriented DRN architecture and showed that customized residual designs effectively predict nonlinear load patterns across many datasets. However, these studies mostly concentrated on aggregate forecasting accuracy and paid little attention to highly variable or high-frequency load dynamics.

To further improve stability and generalization, ensemble-enhanced DRN frameworks were examined. Xu et al.[29] presented an Ensemble Residual Network (ERN) that use snapshot ensemble learning to increase robustness without training many independent models. Chen et al.[30] developed this concept further by including multi-scale feature extraction and snapshot ensembles into a ResNet-based system. Although ensemble-based DRN approaches are superior at generalization, their real-time use may be limited since they typically require greater processing power, lengthier training schedules, and careful snapshot interval design.

Concurrently, learning stability in DRN-based models has been enhanced by data-driven augmentation techniques. In order to capture important load characteristics under a variety of operating settings, Kondaiah et al.[31] designed a Deep-ResNet architecture that emphasizes adaptive feature selection. Transferability across diverse power systems is hampered by susceptibility to changes in data distribution, despite improvements in predicting consistency. Additionally, structural improvements have been implemented to improve feature representation. While Sheng et al.[32] proposed a convolutional residual network (CRN) by redesigning convolutional residual blocks (ResBlocks) to enhance local pattern extraction for high-resolution load data, Ding et al.[33] integrated inception-inspired modules into residual architectures to enable multi-scale learning. These improvements complicate hyperparameter optimization and add more architectural complexity despite increased representational capacity.

To enhance temporal dependency modeling, some studies have integrated DRNs with recurrent and attention-based processes. While Li et al.[34] introduced attention techniques to highlight significant time steps, Tian et al.[35] combined deep residual feature extraction with sequential modeling in a ResNetPlus–LSTM framework. More recently, Sheng et al.[36] introduced a Residual LSTM Plus architecture that tightly connects ResBlocks with recurrent layers. The simultaneous stacking of residual, recurrent, and attention components significantly increases model depth and computational cost, limiting scalability in large-scale or real-time STLF systems, even though these hybrid frameworks often improve forecasting reliability.

Beyond architectural and ensemble-based enhancements, prior DRN-based studies have also explored extensions at the input level to incorporate richer meteorological information. Liu et al.[37] proposed the PCA-DRN framework to extend temperature-based DRN models by integrating multiple weather variables through principal component analysis (PCA). While PCA-DRN effectively reduces input dimensionality and feature redundancy while preserving the residual learning structure, its reliance on linear dimensionality reduction inevitably weakens the physical interpretability of meteorological variables.

In addition to model architecture, the optimization algorithm also plays an important role in determining the convergence behavior and generalization performance of DL models. The theoretical foundation of stochastic optimization can be traced back to stochastic approximation methods, which provide iterative procedures for solving optimization problems under noisy observations[38]. Building on this foundation, stochastic gradient–based optimization approaches have become the dominant training strategy for DNNs. Subsequent developments introduced adaptive gradient–based methods that dynamically adjust learning rates according to historical gradient information, improving optimization efficiency in high-dimensional problems[39]. Among these approaches, the Adam optimizer has been widely adopted in DL due to its ability to combine momentum-based updates with adaptive learning-rate adjustment, enabling stable convergence under stochastic gradient updates[40]. In DRN-based STLF research[27–37], Adam is commonly used to ensure stable training of deep residual architectures. Consequently, employing a consistent optimization strategy provides a controlled setting for investigating the influence of other training factors, such as mini-batch size, on forecasting performance.

Therefore, Adam is adopted in this study due to its ability to provide stable convergence and efficient optimization for deep neural networks with large parameter spaces.

Prior studies in the DL literature have shown that mini-batch size can substantially influence optimization behavior, gradient variance, and generalization performance in neural network training. For example, theoretical and empirical analyses indicate that gradient variance decreases as mini-batch size increases, thereby affecting training trajectories and convergence stability across different model architectures[41]. Small-batch training has also been reported to yield superior generalization performance compared with large-batch regimes on common DL benchmarks[42]. Comprehensive investigations into batch size selection further reveal inherent trade-offs among optimization efficiency, generalization capability, and computational cost[43], while classic studies on large-batch training suggest that excessively large mini-batches may lead to a generalization gap relative to smaller batch sizes[44]. More recent work continues to examine mini-batch size effects under modern adaptive optimization strategies, demonstrating that batch size remains a critical factor even in contemporary DL frameworks[45].

However, within the context of DRN-based STLF, existing research has primarily concentrated on network architecture refinement, ensemble learning mechanisms, and feature-level extensions. In contrast, training configurations are often treated as fixed or secondary design choices. In particular, mini-batch size is typically selected empirically and kept unchanged across experiments, and its influence on forecasting performance and empirical training outcomes is rarely examined in a systematic manner. In practical STLF implementations, mini-batch size is also closely tied to hardware constraints and training efficiency, further highlighting the need for a systematic evaluation of its impact on DRN-based models.

From a training dynamics perspective, this issue is especially relevant for deep residual architectures, where optimization stability and gradient propagation are closely linked to residual learning mechanisms. Variations in mini-batch size may interact with residual connections by altering gradient noise characteristics and update consistency during training. Moreover, given that PCA-DRN extends the DRN framework by incorporating multiple meteorological variables through dimensionality reduction, examining whether similar mini-batch size sensitivity persists in PCA-DRN models is also of practical interest. Accordingly, a systematic empirical investigation of mini-batch size sensitivity is conducted for both DRN and PCA-DRN frameworks in this study.

Methods of research

Description and preprocessing of research data

In real-world power system applications, incomplete records, missing observations, noise contamination, and heterogeneous data formats are commonly encountered, which may significantly degrade forecasting performance if not properly addressed[46]. Therefore, rigorous data preprocessing is a critical prerequisite for improving the robustness and reliability of STLF models. In this study, two real-world load datasets—the New England Independent System Operator (ISO-NE) dataset and the Malaysia Petaling Jaya (MyPJ) dataset—are utilized to investigate STLF performance under different climatic conditions and operational environments.

The ISO-NE dataset contains hourly electricity load and temperature records from March 2003 to December 2006, covering six states in the New England region of the United States: Connecticut, Maine, Massachusetts, New Hampshire, Rhode Island, and Vermont. This dataset represents a temperate climate characterized by pronounced seasonal and interannual demand variability. The load records correspond to system-level electricity demand, while the temperature data represent regional hourly observed temperatures rather than forecasted values. Due to its completeness and standardized structure, the ISO-NE dataset has been widely adopted as a benchmark dataset in load forecasting research. Since the dataset has already undergone standard data quality control and preprocessing by the data provider, no additional cleaning procedures were required in this study.

In contrast, the Malaysia Petaling Jaya (MyPJ) dataset provides a representative case for examining electricity demand patterns in tropical climates. The dataset consists of nationwide hourly electrical 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. Compared with temperate regions, the MyPJ dataset reflects electricity consumption behavior in a tropical climate, where seasonal variations are relatively mild and load dynamics are more strongly influenced by short-term weather conditions.

During the data collection process, missing observations were identified in the raw MyPJ dataset. To preserve temporal continuity and mitigate the impact of missing observations, linear interpolation was applied. The statistical characteristics of the missing data are summarized in Table 1. Given the relatively low missingness rate, linear interpolation provides a practical approach for maintaining temporal continuity while introducing minimal distortion to the underlying load patterns.

As illustrated in Fig. 1, the hourly load profiles of the two datasets exhibit distinct demand characteristics under different climatic environments. The ISO-NE load values generally range from approximately 10,000 to 27,500 megawatts (MW) and exhibit clear seasonal and long-term fluctuations, reflecting the strong influence of winter heating and summer cooling demands typical of temperate regions. In contrast, the MyPJ load values predominantly range between approximately 10,000 and 18,000 MW, demonstrating relatively moderate variability throughout the year. This pattern reflects electricity consumption behavior in tropical climates, where seasonal variation is less pronounced and load dynamics are more strongly affected by short-term weather conditions.

To avoid information leakage during model training and evaluation, data normalization was performed using statistical parameters derived exclusively from the training set, and the same scaling factors were subsequently applied to the corresponding test set. This preprocessing strategy ensures a fair and reliable assessment of forecasting performance under realistic operational scenarios.

Item	Value
Total expected observations	26,304
Number of missing observations	154
Missingness rate	0.59%
Number of days with missing observations	19
Percentage of affected days	1.73%
Longest continuous missing segment	35 h
Average missing sequence length	10.3 h

Table 1. Summary of missing observations in the MyPJ dataset.

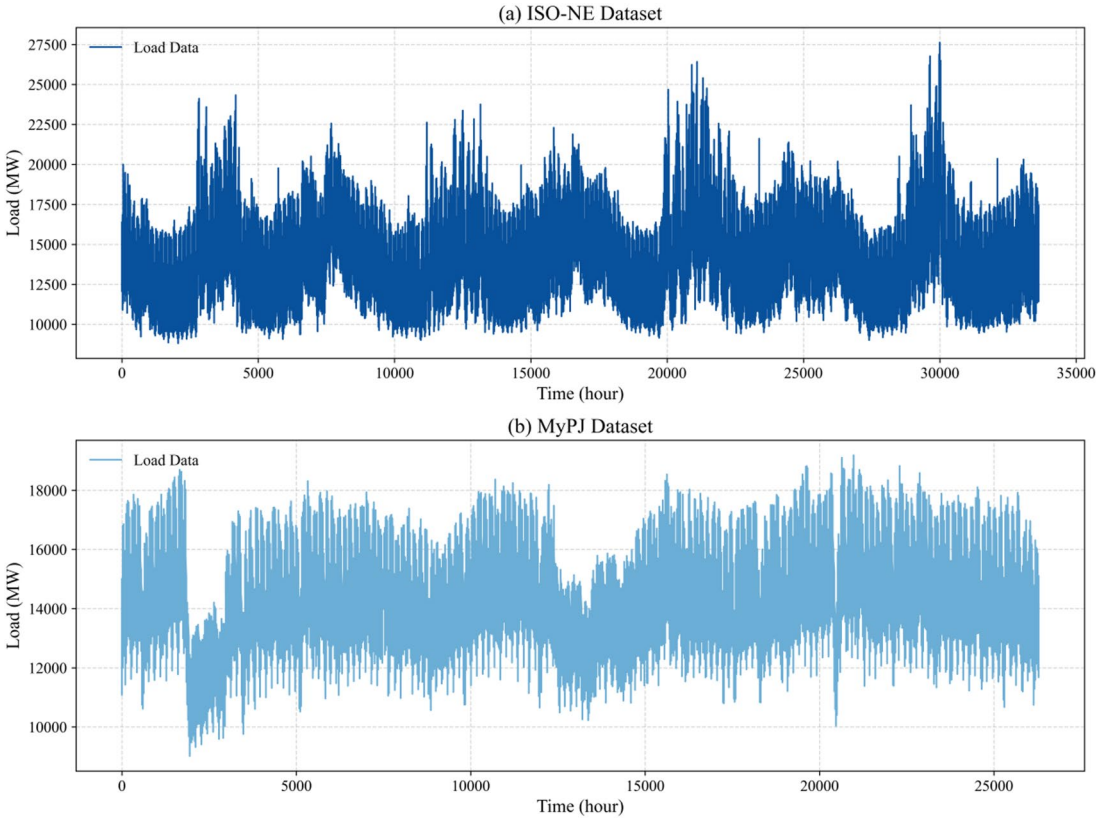

Fig. 1. Hourly load profiles of the two datasets: (**a**) ISO-NE; (**b**) MyPJ.

Model architecture and proposed method

To systematically investigate the influence of mini-batch size on training behavior and forecasting performance, this study adopts two representative residual-based forecasting frameworks: the original DRN[27] and PCA-DRN[37]. Unlike prior DRN-based studies that primarily emphasize architectural refinements to improve predictive accuracy, this work keeps the ResNet structure constant in order to examine the impact of training configuration on empirical training outcomes and generalization performance. These models provide a consistent architectural foundation while enabling comparative analysis across different input representations.

Figure 2 illustrates the architecture of the original DRN model applied to the ISO-NE dataset. Figures 3 and 4 illustrate the DRN and PCA-DRN architectures used for the MyPJ dataset, respectively. The PCA-DRN model is applied only to the MyPJ dataset because it contains multiple meteorological variables that may introduce redundancy in the input representation. In contrast, the ISO-NE dataset provides only a single weather variable, making dimensionality reduction through PCA unnecessary. Therefore, the original DRN framework is adopted for the ISO-NE dataset without PCA-based feature transformation. Despite these differences in input representation, the overall model architecture remains consistent across the two frameworks.

Both DRN and PCA-DRN share the same architecture consisting of two primary components: a basic structure and a ResNetPlus module. The input features include historical load, time-related variables, and meteorological information. The basic structure first processes these inputs to generate preliminary feature representations of electricity demand. Subsequently, the ResNetPlus module—an enhanced version of the traditional ResNet

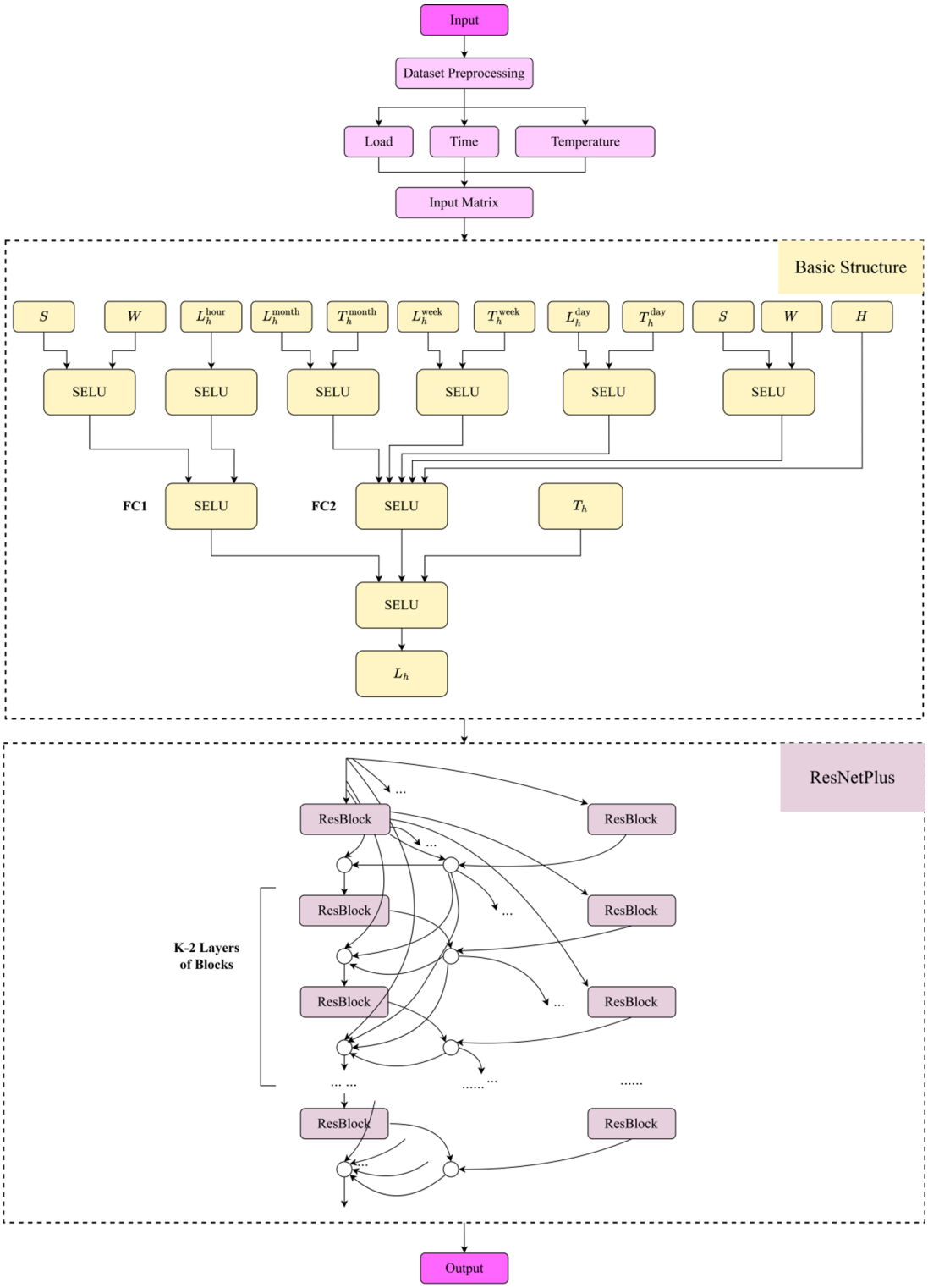

Fig. 2. Architecture of the original DRN framework used for the ISO-NE dataset.

architecture—applies deep residual learning to further refine the learned representations and improve 24-hour ahead forecasting accuracy.

Although the original DRN demonstrates effective forecasting capability, it considers only a limited set of meteorological inputs. In real-world STLF scenarios, electricity demand is often influenced by multiple weather factors that may exhibit strong correlations and redundancy. To address this limitation, the PCA-DRN extends the original DRN framework by incorporating multiple meteorological variables and applying PCA during

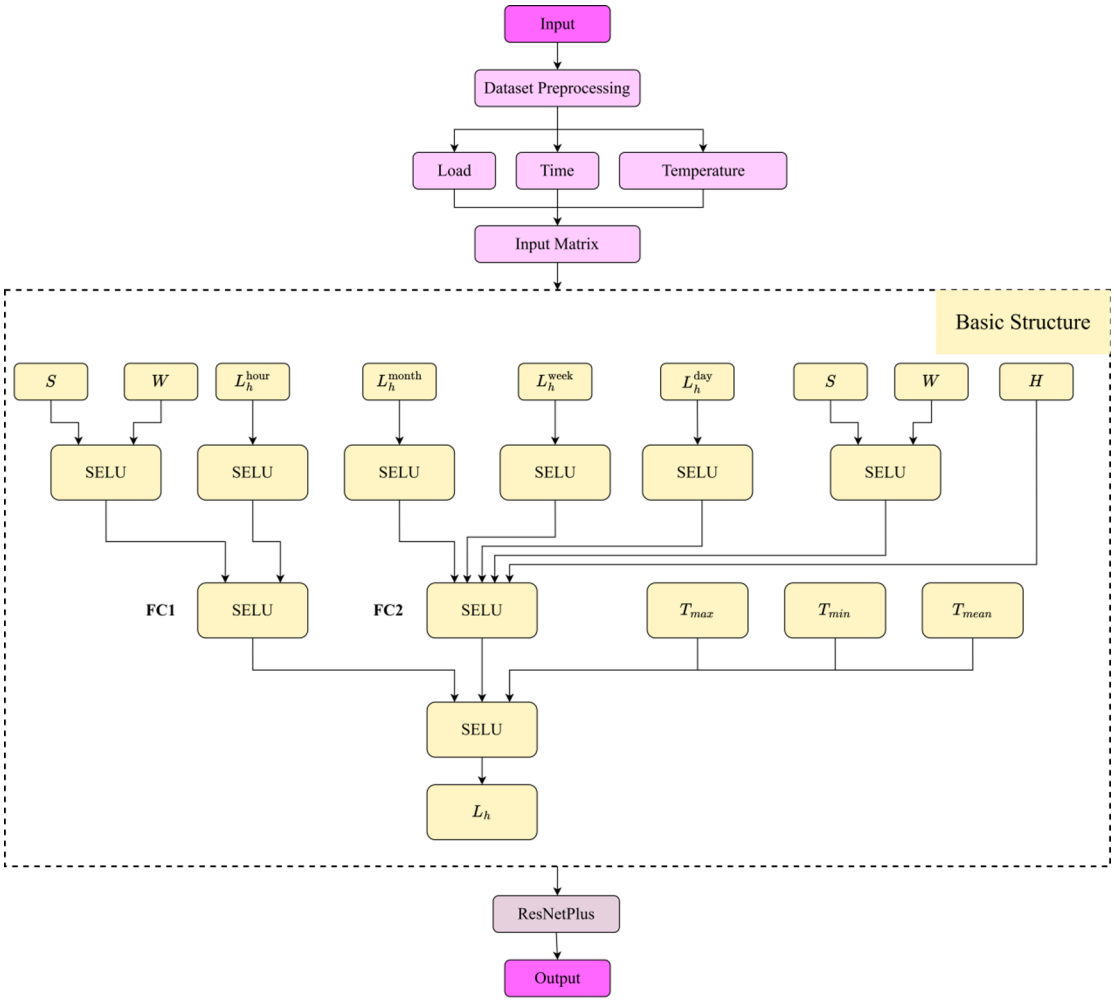

Fig. 3. Architecture of the DRN framework used for the MyPJ dataset.

data preprocessing and feature construction. This process reduces dimensionality and eliminates redundant information while preserving the same two-component residual learning structure.

Both DRN and PCA-DRN adopt a series of hour-specific sub-models for day-ahead forecasting, where each sub-model is responsible for predicting the load of a specific future hour. The outputs of preceding sub-models are iteratively fed back as inputs to capture short-term inter-hour dependencies. The ResNetPlus module then jointly optimizes the resulting 24 hourly forecasts to produce the final day-ahead load profile. All layers in both models—except the output layer—use the Scaled Exponential Linear Unit (SELU) as the activation function.

Within the original DRN framework, day-ahead STLF starts from the first component, known as the basic structure, which is implemented using a series of fully connected (FC) layers to generate preliminary 24-hour-ahead load predictions. This component is designed to capture multi-scale temporal dependencies by integrating historical load information across different time horizons.

For the ISO-NE dataset, the DRN basic structure is configured as follows. 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 the aggregated output 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 $\mathrm{T_h}$ 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. Specifically, S corresponds to the four seasons-spring, summer, autumn, and winter-whereas H includes major public holidays such as Christmas and Independence Day. The aggregated output of this component is denoted as $\mathrm{L_h}$, which is subsequently passed to the ResNetPlus module for further refinement and improvement of forecasting accuracy.

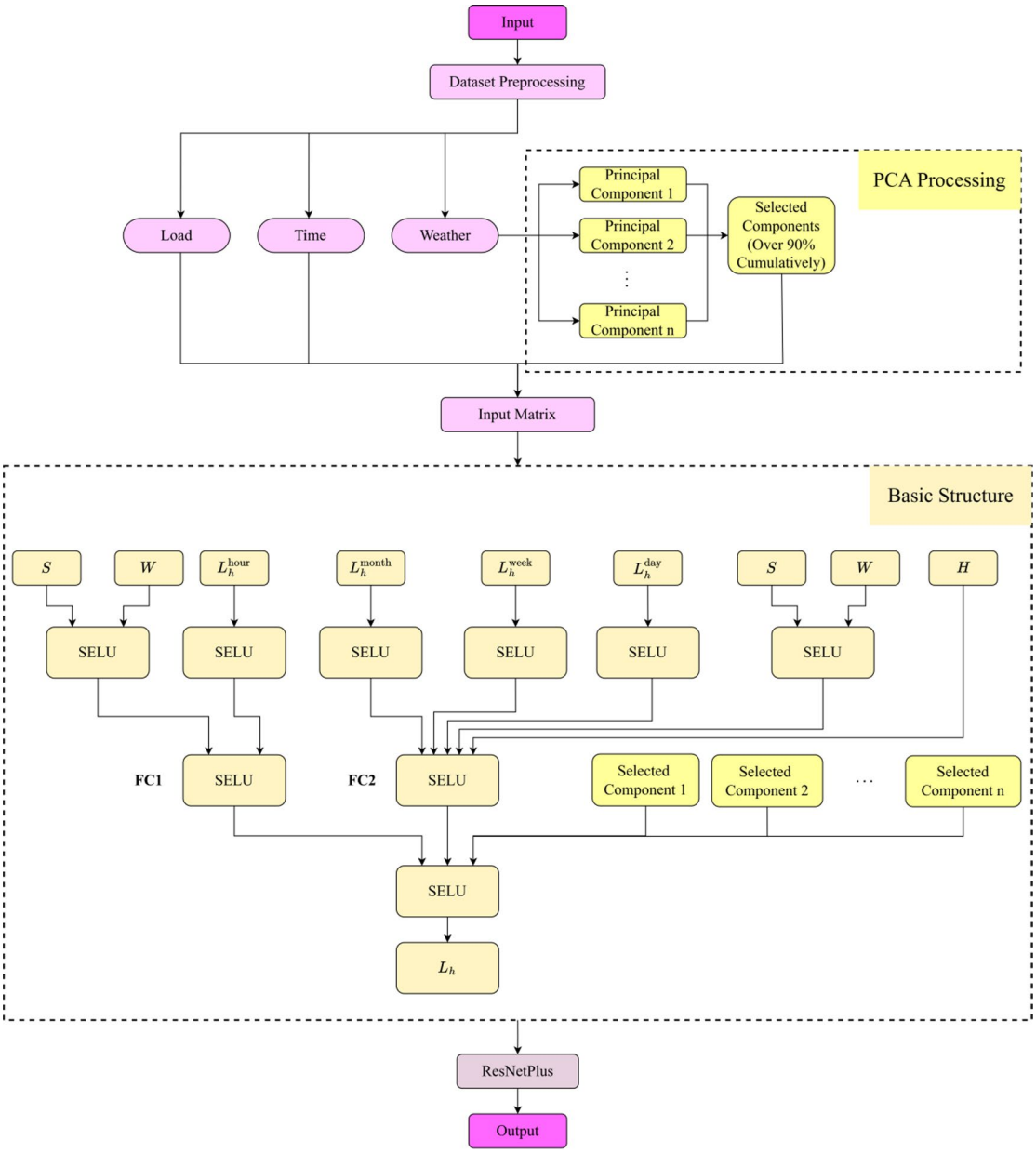

Fig. 4. Architecture of the PCA-DRN framework used for the MyPJ dataset.

For the MyPJ dataset, the same DRN framework is adopted, while the meteorological and categorical variables are defined according to local climatic characteristics. Load observations at the corresponding hour over the most recent 24 -hour period are represented by $\mathrm{L_h^{hour}}$, while daily periodic patterns are described by $\mathrm{L_h^{day}}$, which contains the hourly load values for each day of the previous week. Longer-range temporal dependencies are modeled through $\mathrm{L_h^{week}}$, aggregating load observations at the same hour over the preceding one to eight weeks, and $\mathrm{L_h^{month}}$, which encodes load information from one, two, and three months earlier. Each of these historical load inputs is processed through an individual FC layer with ten hidden neurons. Meteorological effects are incorporated through daily temperature statistics, including mean, maximum, and minimum values ($\mathrm{T_{mean}}$, $\mathrm{T_{max}}$, $\mathrm{T_{min}}$), which are concatenated and transformed by an additional FC layer with ten hidden units. Temporal categorical features, namely season, weekday, and holiday status, are encoded using one-hot representations (S, W and H) and mapped through dedicated FC layers with five hidden neurons, where H includes public holidays such as Eid al-Fitr and Malaysia Independence Day, and S is defined according to local climatic conditions as the rainy and dry seasons. To ensure adequate feature extraction capacity, the intermediate FC layers (FC1 and FC2), as well as the layer immediately preceding the aggregated output, also employ ten hidden neurons. The resulting output is then forwarded to the ResNetPlus module for further refinement and enhancement of overall STLF accuracy.

For the MyPJ dataset, to alleviate the limitations imposed by using a limited set of meteorological inputs, PCA-DRN enhances the original DRN by introducing a PCA-based feature preprocessing scheme prior to

model training. In this framework, the first component, corresponding to the basic structure, preserves the same architectural design as in the original DRN, while the representation of weather information is modified. Instead of directly using raw meteorological variables, PCA is applied to project multiple weather variables into a lowerdimensional feature space, thereby reducing redundancy and retaining the principal components that collectively account for 90% of the cumulative variance. These PCA-derived components are then fed into the basic structure as weather-related inputs. Other elements of the first component remain unchanged, including the FC layers associated with $\mathrm{L_h^{day}}$, $\mathrm{L_h^{week}}$, $\mathrm{L_h^{month}}$ and $\mathrm{L_h^{hour}}$, as well as the one-hot encoded categorical variables S, W and H. Consistent with the original DRN, the aggregated output of the first component, denoted as $\mathrm{L_h}$, is subsequently forwarded to the ResNetPlus module for further refinement.

As the second component shared by both the DRN and PCA-DRN frameworks, ResNetPlus is designed as an enhanced residual learning module that follows the core principles of conventional ResNet architectures while incorporating task-oriented structural refinements. The module consists of multiple stacked building units, each comprising a nonlinear hidden layer with 20 neurons activated by SELU, followed by a linear projection layer that ensures dimensional compatibility for residual aggregation. By repeatedly stacking these units, ResNetPlus forms a deep hierarchical structure capable of progressively refining feature representations. In the adopted configuration, four such units are sequentially grouped to form a ResBlock, and this grouping pattern is replicated across ten hierarchical levels. Shortcut connections are integrated throughout the network to facilitate efficient gradient propagation and stable optimization in deep architectures. Compared with the standard ResNet design, ResNetPlus reorganizes the internal block composition without altering the original hyperparameter settings, thereby enabling more effective utilization of residual learning within the DRN-based forecasting framework.

To balance the structural consistency of daily load profiles with forecasting accuracy, both the original DRN and the PCA-DRN adopt the same unified training objective across all datasets during model optimization. The overall loss function is defined as the sum of two complementary components, reflecting the residual-learning objective of constraining predicted load trajectories within a realistic range while reducing point-wise prediction errors. As shown in Eq. (1), the total loss consists of an error-based term ($\mathrm{Loss_E}$) and a range-aware penalty term ($\mathrm{Loss_R}$). The error term $\mathrm{Loss_E}$, given in Eq. (2), measures forecasting accuracy using the mean absolute percentage error (MAPE), which is computed from the relative difference between the actual normalized load $\mathrm{y_{j,h}}$ and the predicted normalized load $\mathrm{\acute{y}_{j,h}}$ at the h-th hour of the j-th day, where Num denotes the total number of samples and H is fixed at 24 to represent the number of hourly load values per day. Equation (3) defines the range-aware term $\mathrm{Loss_R}$, which penalizes overestimation of daily peak loads and underestimation of trough values by comparing the predicted and observed daily maximum and minimum loads. By jointly considering accuracy and range consistency, this loss formulation enhances the robustness and stability of the learning process in both the original DRN and PCA-DRN frameworks.

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

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

$$\begin{aligned}\mathrm{Loss}_R=&\frac{1}{2Num}\sum_{j=1}^{Num}\max\left(0,\max_h \acute{y}_{(j,h)}-\max_h y_{(j,h)}\right)\\&+\max\left(0,\min_h y_{(j,h)}-\min_h \acute{y}_{(j,h)}\right)\end{aligned} \tag{3}$$

After the model architecture and learning objective have been defined, the practical performance of the model during training remains strongly influenced by the adopted training configurations. In DNNs, these configurations determine the quality of gradient estimation, the frequency of parameter updates, and the overall convergence behavior. Among them, mini-batch size is widely regarded as a key factor affecting training stability and optimization dynamics. Therefore, it is necessary to systematically investigate the impact of mini-batch size on optimization behavior and forecasting performance within the DRN-based STLF framework. In this study, mini-batch size is treated as a core experimental variable rather than a fixed hyperparameter. To ensure fairness and controllability, all experiments strictly maintain identical network architecture, residual learning structure, activation functions, loss function, and optimizer settings, while only the mini-batch size is varied during model training. This design enables an objective evaluation of the effects of mini-batch size on empirical training outcomes and predictive accuracy without interference from architectural or algorithmic confounding factors.

From the perspective of optimization theory and empirical DL practice, mini-batch size directly influences gradient variance and parameter update behavior. Smaller mini-batches tend to introduce higher stochastic noise into gradient estimates, which may encourage exploration of flatter minima and thus improve generalization, albeit often at the expense of slower convergence and increased training instability. In contrast, larger mini-batches typically provide more stable gradient estimates and higher computational efficiency, but may also reduce beneficial gradient noise and lead to inferior generalization performance. Given the depth and explicit residual connections of DRN-based architectures, systematically examining the interaction between different mini-batch size regimes and residual learning mechanisms is of particular relevance in STLF. For analytical convenience, the investigated mini-batch sizes are conceptually grouped into small-, medium-, and large-scale

training regimes in the discussion to characterize optimization behavior and performance trends across different training scales.

It should be emphasized that the same mini-batch size variation strategy is consistently applied to both the original DRN and the PCA-DRN frameworks. Although PCA-DRN extends the original DRN by incorporating dimensionally reduced meteorological features through PCA, its residual learning structure, training objective, and optimization pipeline remain unchanged. This parallel experimental design allows for a direct comparison of mini-batch size sensitivity under different input representations, thereby revealing whether PCA-based feature compression alters the interaction between batch size, optimization dynamics, and forecasting performance. By maintaining architectural consistency and systematically varying the mini-batch size across both residual-based models, the proposed methodology provides a reproducible and comprehensive framework for analyzing the effects of training configurations in DRN-based STLF. Detailed experimental settings, including the specific mini-batch size values and other training parameters, are presented in the subsequent section.

Design of experimentation

This work uses real-world power load datasets from both ISO-NE and MyPJ. For the ISO-NE dataset, the model was trained using observations from March 2003 to December 2005, while data from 2006 was held aside for testing, yielding 24,888 training samples and 8760 testing samples in total. For the MyPJ dataset from Malaysia, the model was trained using observations from January 2020 to December 2021, while data from 2022 was held aside for testing, yielding 17,544 training samples and 8760 testing samples in total. The datasets cover a variety of load fluctuations caused by meteorological effects and provide a solid foundation for evaluating forecasting effectiveness under both temperate and tropical climatic conditions.

Both the DRN and PCA-DRN frameworks are implemented using the default architectural configurations established in prior studies, without introducing any structural modifications. To systematically evaluate the sensitivity of model performance to training batch size, this study focuses on the role of mini-batch size as a key training configuration parameter. The mini-batch size determines the number of training samples used to estimate the gradient in each parameter update during optimization, thereby influencing gradient stochasticity and the empirical generalization performance of the model. Based on this consideration, a set of representative mini-batch sizes covering a wide range of training scales is examined, specifically 8, 16, 32, 64, 128, 256, and 512. For analytical convenience, these batch sizes are descriptively grouped into small-, medium-, and large-scale training regimes, where 8, 16, and 32 correspond to small-batch training, 64 and 128 represent medium-batch training, and 256 and 512 are treated as large-batch training; this grouping is adopted solely to facilitate the discussion of optimization behavior and performance trends across different training scales rather than to define a universal categorization of mini-batch size. To ensure fairness and reproducibility, all other training configurations—including network architecture, loss function, optimizer type, learning rate, and number of training epochs—are kept strictly identical across different mini-batch size settings, such that any observed differences in forecasting accuracy can be primarily attributed to the effect of mini-batch size.

To provide representative benchmark references for comparison, several DL models that have been widely adopted in STLF studies are selected as baseline methods. These baselines include convolution-based CNN and TCN models, the recurrent-based LSTM model, and the attention-driven Transformer model, as well as two representative hybrid architectures, namely Transformer–LSTM[24] and CNN–LSTM–MMA[25], which reflect typical applications of different modeling paradigms in STLF tasks. The inclusion of these baseline models aims to evaluate the forecasting performance of the DRN and PCA-DRN frameworks under their respective optimal mini-batch size configurations, rather than to conduct a comprehensive architectural competition among different models. To ensure fairness and consistency in the comparative analysis, all baseline models are trained and evaluated using the same mini-batch size as that adopted by the corresponding DRN or PCA-DRN under its optimal configuration, while all other training conditions are kept identical. By conducting the comparison under a unified training scale and experimental protocol, this design enables an objective assessment of the relative forecasting performance of residual-network-based models against mainstream DL approaches and facilitates a clearer analysis of the practical impact of mini-batch size optimization on DRN-based STLF.

For consistency across different methods, the CNN baseline adopts a one-dimensional convolution (Conv1D) architecture with 32 filters, a kernel size of 3, Rectified Linear Unit (ReLU) activation, and He normal initialization, while all remaining settings follow the default configuration. The TCN model is implemented using stacked Conv1D layers with dilated causal convolutions to capture temporal dependencies across multiple time scales, where the dilation rates are set to {1, 2, 4, 8}; the Conv1D layers in the TCN employ the same number of filters, kernel size, and activation function as the CNN baseline, with all other parameters kept unchanged. For the LSTM baseline, the number of hidden units is set to 64, and all remaining architectural and training configurations follow default settings. The Transformer baseline adopts a standard encoder-only architecture, with the embedding dimension set to 64 to provide sufficient modeling capacity for self-attention mechanisms without constraining it to convolutional filter sizes; the model consists of a single encoder layer with eight attention heads and a feedforward network of dimension 2048, while all other components, including positional encoding and dropout (set to 0.1), follow default configurations. The Transformer–LSTM and CNN–LSTM–MMA hybrid baselines are implemented strictly according to the architectures and parameter settings reported in their original studies.

Based on the previously introduced loss function, the model is trained for a total of 700 epochs, consisting of an initial training stage of 600 epochs followed by two short training phases of 50 epochs each[27]. To mitigate overfitting and enhance model robustness, a snapshot ensemble learning strategy is adopted, in which the model weights are preserved at the end of each 50-epoch phase and the final predictions are obtained by averaging the outputs from multiple snapshots[47]. In addition to providing a computationally efficient alternative to multiple independent training runs, snapshot averaging improves prediction stability and generalization performance[48].

Model training is conducted using the Adam[49] optimizer, which has been widely adopted in previous DRN-based STLF studies. Using a consistent optimizer across experiments allows the present study to isolate the influence of mini-batch size on forecasting performance.

To rigorously evaluate the statistical significance of the observed performance differences, a nonparametric bootstrap resampling procedure with 10,000 iterations is employed. Unlike the paired Student's t-test, which assumes normally distributed paired differences, the bootstrap method is distribution-free and therefore provides a more robust framework for model comparison[50]. In this study, the bootstrap procedure is applied to the difference in absolute percentage errors between competing models across all prediction points. For each prediction point, the absolute percentage error is computed using the predicted and observed load values, and the difference in errors between the two models is obtained. Bootstrap resampling with replacement is then applied to this error-difference series to generate an empirical distribution of the mean performance difference. Statistical significance is determined based on two complementary criteria. First, an improvement is considered statistically significant if the 95% confidence interval (CI) of the mean performance difference lies entirely above zero, whereas the difference is regarded as insignificant if the interval includes zero. Second, a bootstrap-derived p-value smaller than 0.05 indicates statistical significance at the 95% confidence level. It should be noted that a reported value of $p \approx 0$ denotes an extremely small probability (typically < 0.0001) rather than an exact zero.

TensorFlow 2.10.0 and Keras 2.10.0 served as the DL backends for all experiments, which were carried out in a Python 3.8 environment. A Lenovo laptop 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) was used for the calculations.

Metrics for evaluation

In line with prior DRN-based research on STLF, this study evaluates forecasting performance using multiple complementary metrics, among which MAPE is selected as the primary evaluation criterion due to its intuitive interpretability and widespread adoption in the STLF literature, and it serves as the main basis for model comparison and subsequent statistical analysis. To provide a more comprehensive assessment, additional error- and correlation-based indicators are also considered, including Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Normalized Mean Squared Error (NMSE), the Correlation Coefficient (R), and the Coefficient of Determination (R^2). Specifically, MAPE and MAE are treated as point-forecast error metrics, measuring average relative and absolute deviations at the sample level, respectively. MSE, RMSE, and NMSE are squared-error metrics that penalize larger deviations more heavily, thereby reflecting error dispersion and sensitivity to large forecast errors. R and R^2 quantify overall agreement and goodness-of-fit between predicted and observed load profiles, complementing error-based measures from a fitting-consistency perspective. In Eqs. (4)–(10), $\hat{y}_i$ and $\hat{y}_i$ denote the actual and predicted values of the i-th sample, respectively, while $\bar{y}$, $\bar{\hat{y}}$ and σ_y^2 represent the mean values of the corresponding series and the variance of the observed data; N indicates the total number of samples, and analogous notation applies to the remaining metrics. In general, lower values of MAPE, MAE, MSE, RMSE, and NMSE indicate smaller forecasting errors and improved generalization, whereas higher R and R^2 values imply stronger correlation and better model fitting.

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

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

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

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

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

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

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

Batch size	MAPE	MAE	MSE	RMSE	NMSE	R	R^2
8	0.017522	0.011186	0.000316	0.017790	0.020999	0.989625	0.979001
16	0.017290	0.011209	0.000332	0.018218	0.022023	0.988990	0.977977
32	0.017182	0.011138	0.000308	0.017548	0.020432	0.989767	0.979568
64	0.016695	0.010805	0.000306	0.017502	0.020326	0.989806	0.979674
128	0.020716	0.013228	0.000397	0.019932	0.026360	0.987876	0.973640
256	0.021941	0.014209	0.000482	0.021965	0.032011	0.984289	0.967989
512	0.025728	0.016785	0.000668	0.025841	0.044309	0.978411	0.955691

Table 2. Numerical performance of DRN under different mini-batch sizes on ISO-NE dataset.

Batch size	MAPE	MAE	MSE	RMSE	NMSE	R	R^2
8	0.057450	0.030362	0.002820	0.053103	0.099047	0.949963	0.900953
16	0.054908	0.028872	0.002356	0.048535	0.082741	0.958931	0.917259
32	0.052514	0.026467	0.002050	0.045278	0.072007	0.964032	0.927993
64	0.050746	0.025463	0.002025	0.045004	0.071139	0.964017	0.928861
128	0.055581	0.028492	0.002254	0.047473	0.079158	0.961069	0.920842
256	0.058254	0.030270	0.002335	0.048324	0.082022	0.960204	0.917978
512	0.060499	0.030848	0.002438	0.049381	0.085648	0.957420	0.914352

Table 3. Numerical performance of DRN under different mini-batch sizes on MyPJ dataset.

Experimental evaluation and discussion

Effect of mini-batch size on the performance of DRN on ISO-NE dataset

Table 2 summarizes the forecasting performance of the DRN model under different mini-batch size configurations on the ISO-NE dataset. The experimental results reveal a clear dependence of forecasting accuracy on the choice of mini-batch size, with medium batch sizes providing the most favorable performance.

When trained with small mini-batch sizes, the DRN exhibits relatively limited forecasting accuracy. At a batch size of 8, the model achieves a MAPE of 0.017522, which is higher than those obtained with moderate batch sizes. Although small-batch training introduces stochastic gradient noise that may facilitate exploration during optimization, excessive stochasticity can reduce the stability of parameter updates and thus hinder the convergence of deep residual architectures.

As the mini-batch size increases to 16 and 32, the forecasting accuracy improves gradually. In particular, the MAPE decreases from 0.017290 at batch size 16 to 0.017182 at batch size 32, indicating improved point prediction accuracy under moderately sized batches. This improvement suggests that moderate batch sizes provide more reliable gradient estimates while still preserving sufficient stochasticity to avoid premature convergence.

The best overall performance is achieved at a mini-batch size of 64, where the DRN obtains the lowest MAPE value of 0.016695. Compared with batch size 32, the improvement at batch size 64 is accompanied by reductions in MAE, MSE, RMSE, and NMSE, together with slightly improved correlation metrics (R and R^2). These results indicate that a medium batch size provides a more stable optimization process while maintaining sufficient gradient variability for effective generalization.

However, when the mini-batch size is further increased to 128, 256, and 512, forecasting performance deteriorates noticeably. The MAPE increases to 0.020716 at batch size 128 and continues to rise as the batch size becomes larger. Similar degradation trends are observed in MAE, RMSE, and NMSE, while correlation-based metrics also decline accordingly. This pattern suggests that excessively large mini-batches reduce beneficial gradient noise and may guide the optimization process toward sharper minima, thereby weakening the generalization capability of the model.

Taken together, the DRN demonstrates a clear optimal mini-batch size regime on the ISO-NE dataset, with batch size 64 yielding the best forecasting performance and batch size 32 providing the second-best results.

Effect of mini-batch size on the performance of DRN and PCA-DRN on MyPJ dataset

DRN

Table 3 reports the forecasting performance of the original DRN under different mini-batch size configurations. The experimental results reveal a clear batch-size-dependent performance pattern, with medium batch sizes yielding superior forecasting accuracy.

When trained with small mini-batch sizes, the DRN exhibits relatively inferior performance. At a batch size of 8, the model records the highest MAPE (0.057450), indicating limited point-wise forecasting accuracy under highly stochastic gradient updates. Although small-batch training introduces gradient noise that may facilitate exploration during optimization, excessive stochasticity appears to hinder stable convergence in the deep residual architecture.

As the mini-batch size increases to 16 and 32, a consistent improvement in forecasting accuracy is observed. In particular, the MAPE decreases from 0.054908 at batch size 16 to 0.052514 at batch size 32, indicating

progressively improved point prediction accuracy. This improvement suggests that moderate batch sizes enable a more reliable estimation of gradient directions while retaining sufficient stochasticity to avoid premature convergence.

The best overall performance of the DRN is achieved at a mini-batch size of 64, where the lowest MAPE value of 0.050746 is obtained. Compared with batch size 32, the performance gain at batch size 64 is modest but consistent, indicating enhanced convergence stability without sacrificing generalization. Supporting evidence from additional metrics shows simultaneous reductions in MAE, RMSE, and NMSE, as well as improved correlation consistency (R and R^2), confirming that the improvement is not limited to point-wise accuracy but also reflected in reduced error dispersion and better overall fitting.

However, further increasing the mini-batch size beyond 64 leads to a gradual deterioration in forecasting performance. At batch sizes of 128, 256, and 512, the MAPE increases steadily, accompanied by unfavorable changes in squared-error-based and correlation-based metrics. This trend suggests that excessively large mini-batches reduce beneficial gradient noise and may bias optimization toward sharp minima, resulting in inferior generalization.

Taken together, the DRN demonstrates a clear optimal mini-batch size regime on the MyPJ dataset, with batch size 64 yielding the best forecasting performance and batch size 32 providing the second-best results.

Dimensionality reduction of meteorological variables for PCA-DRN modeling

To investigate the underlying variance structure of the meteorological variables in the MyPJ dataset and to mitigate feature redundancy, PCA was applied prior to model training. A cumulative explained variance threshold of 90% was adopted as the criterion for component retention, resulting in the selection of five principal components. These five components jointly account for 94.12% of the total variance in the original meteorological feature set, indicating that the dominant information is preserved despite substantial dimensionality reduction. Specifically, PCA component 1 explains 37.88% of the total variance, while PCA component 2, PCA component 3, PCA component 4, and PCA component 5 explain 19.62%, 14.23%, 12.08%, and 8.54% of the variance, respectively. These results demonstrate that the retained components provide a compact and low-redundancy representation of the primary variability inherent in the meteorological data.

The relative contribution of each meteorological variable to the retained principal components is quantified by their loading values. For PCA component 1, the loadings of rainfall, mean temperature, minimum temperature, maximum temperature, mean wind speed, maximum wind speed, and maximum wind direction are 0.326146, −0.579888, −0.503639, −0.442409, −0.293040, 0.105115, and 0.105159, respectively, indicating that this component is primarily associated with overall temperature variability. In PCA component 2, the corresponding loadings are 0.442154, 0.082323, −0.203754, 0.365551, 0.318846, 0.701837, and −0.168348, suggesting dominant influences from wind intensity and precipitation-related factors. PCA component 3 exhibits loadings of 0.099971, −0.091294, −0.011570, −0.130403, 0.079191, −0.250668, and −0.946270, with maximum wind direction contributing the largest absolute loading. The loadings of PCA component 4 are 0.381735, 0.141974, 0.147226, 0.413782, −0.792360, −0.087986, and −0.075194, reflecting combined effects of wind speed, temperature, and precipitation. For PCA component 5, the loadings are 0.709699, 0.086094, 0.123478, −0.051551, 0.387480, −0.512842, and 0.240546, highlighting coupled rainfall–wind behavior under varying wind strength conditions. Overall, these principal components preserve the physical interpretability of the meteorological variables while compressing the original feature space into a low-dimensional, orthogonal, and information-concentrated representation, thereby providing a rational and stable meteorological input basis for subsequent analyses of training behavior and forecasting performance under different mini-batch size configurations in the PCA-DRN model.

PCA-DRN

Table 4 summarizes the forecasting performance of the PCA-DRN under the same mini-batch size configurations. The experimental results reveal a clear batch-size-dependent performance pattern, with medium batch sizes providing the most favorable forecasting accuracy.

When trained with small mini-batch sizes, PCA-DRN exhibits relatively limited forecasting accuracy. At a batch size of 8, the MAPE reaches 0.059948, indicating that excessive gradient stochasticity adversely affects convergence stability even when the input dimensionality is reduced through PCA-based feature compression.

As the mini-batch size increases to 16 and 32, substantial improvements in forecasting accuracy are observed. The MAPE decreases from 0.052160 at batch size 16 to 0.049994 at batch size 32, demonstrating enhanced point-wise prediction accuracy under moderately sized batches. This trend suggests that PCA-DRN also benefits from a balanced optimization regime in which gradient estimates become more stable while maintaining sufficient stochasticity to support effective generalization.

The best overall performance is achieved at a mini-batch size of 64, where the PCA-DRN obtains the lowest MAPE value of 0.048943. Compared with batch size 32, the improvement at batch size 64 is consistently reflected across all evaluation metrics, including MAE, RMSE, and NMSE, and is accompanied by the highest values of R and R^2. These results indicate that medium batch sizes not only improve point prediction accuracy but also enhance the overall consistency between predicted and observed load profiles.

However, when the mini-batch size is further increased to 128, 256, and 512, forecasting accuracy gradually deteriorates. The MAPE increases steadily, while correlation-based metrics decline accordingly. This pattern suggests that excessively large mini-batches reduce beneficial gradient noise and may guide the optimization process toward sharper minima, thereby weakening the generalization capability of the model.

Taken together, the PCA-DRN exhibits the same optimal mini-batch size regime as the original DRN, with batch size 64 achieving the best forecasting performance and batch size 32 providing the second-best results.

Batch size	MAPE	MAE	MSE	RMSE	NMSE	R	R^2
8	0.059948	0.031435	0.002760	0.052540	0.096957	0.952112	0.903043
16	0.052160	0.027400	0.002273	0.047673	0.079828	0.959873	0.920172
32	0.049994	0.026254	0.001866	0.043193	0.065527	0.967339	0.934473
64	0.048943	0.024916	0.001830	0.042777	0.064271	0.968054	0.935729
128	0.051480	0.025990	0.002094	0.045765	0.073565	0.962609	0.926435
256	0.054734	0.028104	0.002274	0.047688	0.079876	0.959458	0.920124
512	0.057109	0.028586	0.002356	0.048538	0.082750	0.958484	0.917250

Table 4. Numerical performance of PCA-DRN under different mini-batch sizes on MyPJ dataset.

Model	MAPE	MAE	MSE	RMSE	NMSE	R	R^2
CNN	0.020799	0.013338	0.000416	0.020389	0.027582	0.986180	0.972418
TCN	0.018499	0.011886	0.000336	0.018335	0.022305	0.988860	0.977695
LSTM	0.021271	0.013687	0.000436	0.020884	0.028940	0.985697	0.971060
Transformer	0.023984	0.015733	0.000620	0.024894	0.041120	0.979579	0.958880
Transformer-LSTM	0.019347	0.012521	0.000382	0.019551	0.025364	0.987302	0.974636
CNN-LSTM-MMA	0.021194	0.013759	0.000457	0.021382	0.030335	0.985131	0.969665
DRN	0.016695	0.010805	0.000306	0.017502	0.020326	0.989806	0.979674

Table 5. Comparative forecasting performance of different models on the ISO-NE dataset under the optimal mini-batch size configuration.

Comparative performance analysis with baseline models

ISO-NE dataset

To further evaluate the effectiveness of the residual learning framework, the forecasting performance of the DRN is compared with several representative DL baseline models on the ISO-NE dataset. All models are trained using the same mini-batch size configuration (batch size = 64) to ensure a fair comparison. This unified training setup allows the performance differences to primarily reflect variations in model architecture rather than discrepancies caused by training hyperparameters.

Table 5 presents the comparative forecasting results of all considered models, including convolution-based architectures (CNN and TCN), recurrent models (LSTM), attention-based models (Transformer), hybrid architectures (Transformer–LSTM and CNN–LSTM–MMA), and the residual-based DRN model.

Among the baseline models, TCN achieves the best performance, with a MAPE of 0.018499, indicating that temporal convolutional structures can effectively capture short-term temporal dependencies in electricity load data. The hybrid Transformer–LSTM model also demonstrates relatively competitive performance, achieving a MAPE of 0.019347, which reflects the benefit of combining attention mechanisms with sequential modeling.

However, the DRN consistently outperforms all baseline models, achieving the lowest MAPE of 0.016695. In addition to improved MAPE, the DRN also records the lowest values for MAE, MSE, RMSE, and NMSE, while simultaneously achieving the highest R (0.989806) and R^2 (0.979674) values among all compared models. These results indicate that the residual learning mechanism effectively stabilizes gradient propagation in deep architectures and enables the model to capture complex nonlinear load patterns more accurately.

These findings indicate that residual-network-based forecasting provides superior predictive performance compared with mainstream DL architectures on the ISO-NE dataset. The improved accuracy and stronger correlation metrics suggest that the DRN is particularly effective for modeling the complex load dynamics observed in temperate-climate electricity systems.

MyPJ dataset

To further assess the effectiveness of residual-based forecasting models in a tropical electricity system, the forecasting performance of DRN and PCA-DRN is compared with several representative DL baseline models on the MyPJ dataset. All models are trained under the same mini-batch size configuration (batch size = 64) to ensure consistency in the training process and to allow architectural differences to be clearly reflected in the forecasting results.

Table 6 summarizes the numerical forecasting results of all considered models, including convolution-based models (CNN and TCN), recurrent models (LSTM), attention-driven models (Transformer), hybrid architectures (Transformer–LSTM and CNN–LSTM–MMA), and residual-based models (DRN and PCA-DRN).

Among the baseline methods, CNN–LSTM–MMA achieves the best performance among conventional baseline models, with a MAPE of 0.054391, indicating that combining convolutional feature extraction with sequential modeling and attention-based fusion can improve predictive accuracy. In contrast, purely convolutional models (CNN and TCN) and recurrent models (LSTM) exhibit relatively higher forecasting errors, while the Transformer-based architectures show comparatively inferior performance under the adopted training configuration, likely due to their sensitivity to data scale and optimization settings.

Model	MAPE	MAE	MSE	RMSE	NMSE	R	R^2
CNN	0.053767	0.026603	0.002291	0.047865	0.080470	0.959310	0.919530
TCN	0.058103	0.030002	0.002284	0.047789	0.080217	0.959847	0.919783
LSTM	0.054929	0.029391	0.002264	0.047583	0.079524	0.961677	0.920476
Transformer	0.059608	0.028453	0.002584	0.050832	0.090756	0.956054	0.909244
Transformer-LSTM	0.060019	0.032521	0.002564	0.050632	0.090045	0.959019	0.909955
CNN-LSTM-MMA	0.054391	0.028353	0.002105	0.045883	0.073944	0.963565	0.926056
DRN	0.051082	0.025190	0.002212	0.047033	0.077698	0.960669	0.922302
PCA-DRN	0.048943	0.024916	0.001830	0.042777	0.064271	0.968054	0.935729

Table 6. Comparative forecasting performance of different models on the MyPJ dataset under the optimal mini-batch size configuration.

1st Model	2nd Model	MAPE ± SD (1st model)	MAPE ± SD (2nd model)	Mean difference	CI (95%)	p-value
DRN (Batch Size = 32)	DRN (Batch Size = 64)	0.017182 ± 0.019726	0.016695 ± 0.020136	0.000488	[0.000208, 0.000767]	0.000400

Table 7. Bootstrap distribution of MAPE differences on the ISO-NE dataset.

1st Model	2nd Model	MAPE ± SD (1st model)	MAPE ± SD (2nd model)	Mean difference	CI (95%)	p-value
DRN (Batch Size = 32)	DRN (Batch Size = 64)	0.052514 ± 0.106031	0.050746 ± 0.109725	0.001768	[0.001172, 0.002342]	≈ 0
PCA-DRN (Batch Size = 32)	PCA-DRN (Batch Size = 64)	0.049994 ± 0.088990	0.048943 ± 0.096670	0.001051	[0.000288, 0.001813]	0.007200
DRN (Batch Size = 32)	PCA-DRN (Batch Size = 32)	0.052514 ± 0.106031	0.049994 ± 0.088990	0.002520	[0.001422, 0.003650]	≈ 0
DRN (Batch Size = 64)	PCA-DRN (Batch Size = 64)	0.050746 ± 0.109725	0.048943 ± 0.096670	0.001802	[0.000794, 0.002813]	0.000400

Table 8. Bootstrap distribution of MAPE differences for DRN and PCA-DRN on the MyPJ dataset.

The DRN demonstrates improved performance compared with all conventional baseline models, achieving a MAPE of 0.051082. This improvement highlights the effectiveness of residual learning in enhancing optimization stability and improving predictive capability when modeling complex nonlinear load patterns. Supporting metrics, including MAE, NMSE, and correlation-based indicators, further confirm the improved fitting consistency of the DRN relative to non-residual baseline architectures.

The PCA-DRN achieves the best forecasting performance among all compared models. Under the mini-batch size of 64, PCA-DRN records the lowest MAPE (0.048943), MAE (0.024916), MSE (0.001830), RMSE (0.042777), and NMSE (0.064271), together with the highest correlation coefficient ($R = 0.968054$) and coefficient of determination ($R^2 = 0.935729$). Compared with the original DRN, PCA-DRN further reduces the MAPE by approximately 4.2%, indicating that PCA-based dimensionality reduction effectively removes redundancy among meteorological variables and improves forecasting accuracy without altering the residual learning structure.

These findings demonstrate that residual-network-based models consistently outperform mainstream DL approaches under a unified training configuration. In addition, the superior performance of PCA-DRN indicates that incorporating multiple meteorological variables through principled dimensionality reduction can further enhance forecasting accuracy while maintaining stable model training. This comparative evaluation provides representative performance references that highlight the practical advantages of residual-network-based frameworks for STLF in tropical environments.

Statistical significance assessment using bootstrap resampling

To rigorously evaluate whether the observed differences in forecasting accuracy are statistically meaningful rather than incidental, a nonparametric Bootstrap resampling procedure is employed to assess the significance of MAPE differences between selected model configurations. Tables 7 and 8 summarize the Bootstrap-based statistical comparison results for the ISO-NE and MyPJ datasets, respectively. The reported statistics include the mean and standard deviation (SD) of the MAPE distribution for each model configuration, the mean difference in MAPE between paired models, the corresponding 95% CI, and the associated p-value. In these comparisons, a positive mean difference indicates that the second model listed in each pair achieves lower forecasting error.

First, the effect of mini-batch size on forecasting accuracy is examined by comparing batch sizes 32 and 64 for both residual-based models. On the ISO-NE dataset, increasing the mini-batch size from 32 to 64 leads to a positive mean MAPE difference of 0.000488, with the 95% CI entirely above zero ([0.000208, 0.000767]) and a p-value of 0.000400. This result indicates that the improvement obtained with batch size 64 is statistically significant, confirming that the observed performance gain is unlikely to be caused by random sampling variability.

A similar pattern is observed on the MyPJ dataset. For the original DRN, increasing the mini-batch size from 32 to 64 results in a mean MAPE difference of 0.001768, with a 95% CI of [0.001172, 0.002342] and a p-value

close to zero. Likewise, for PCA-DRN, the comparison between batch sizes 32 and 64 yields a mean MAPE difference of 0.001051 with a 95% CI of [0.000288, 0.001813] and a p-value of 0.007200. Since the confidence intervals for both models lie entirely above zero, the improvement achieved by adopting batch size 64 can be considered statistically significant. These results provide strong statistical evidence supporting the empirical observation that medium-scale mini-batch training yields the most favorable forecasting performance.

Subsequently, cross-model comparisons are conducted to evaluate whether PCA-DRN consistently outperforms the original DRN under identical batch size settings. At batch size 32, PCA-DRN achieves a lower MAPE than DRN, with a mean difference of 0.002520 and a 95% CI of [0.001422, 0.003650], accompanied by a p-value close to zero. This result indicates a statistically significant advantage of PCA-DRN over DRN under the same moderate batch size regime.

The superiority of PCA-DRN is further confirmed under the optimal batch size configuration of 64. In this case, the mean MAPE difference between DRN and PCA-DRN is 0.001802, with a 95% CI of [0.000794, 0.002813] and a p-value of 0.000400. The confidence interval again lies entirely above zero, demonstrating that PCA-DRN maintains a statistically significant performance advantage even when both models are trained under their respective optimal mini-batch size settings.

Overall, the Bootstrap-based statistical analysis validates two key empirical findings. First, the improvements in forecasting accuracy obtained by increasing the mini-batch size from 32 to 64 are statistically significant for both DRN and PCA-DRN, confirming the existence of an optimal mini-batch size regime. Second, PCA-DRN consistently and significantly outperforms the original DRN under identical batch size configurations, indicating that PCA-based meteorological feature compression provides reliable and reproducible improvements in forecasting accuracy rather than marginal or incidental gains.

Summary

This section provides a comprehensive empirical evaluation of the effects of mini-batch size on the forecasting performance of the DRN and PCA-DRN frameworks, integrating numerical performance analysis, comparative evaluation with baseline models, and statistical significance assessment.

First, the intra-model analysis reveals that both DRN and PCA-DRN exhibit clear sensitivity to the selection of mini-batch size. Across the examined configurations, a consistent performance pattern is observed in which medium-scale mini-batch training achieves a more favorable balance between optimization stability and generalization capability. Very small mini-batch sizes introduce excessive gradient stochasticity that may destabilize convergence, whereas overly large mini-batches lead to deteriorated forecasting accuracy due to reduced gradient noise. Among the investigated configurations, a mini-batch size of 64 yields the best overall forecasting performance for both residual-based models, while a batch size of 32 provides the second-best results.

Second, a direct comparison between DRN and PCA-DRN under identical mini-batch settings indicates that PCA-DRN consistently achieves lower forecasting errors across all batch sizes. This observation suggests that incorporating multiple meteorological variables through PCA-based dimensionality reduction effectively reduces feature redundancy and improves predictive accuracy while preserving the residual learning structure of the original DRN.

Third, comparative experiments with representative DL baseline models further demonstrate the effectiveness of residual-based forecasting frameworks. Under a unified training configuration, both DRN and PCA-DRN outperform conventional convolutional, recurrent, attention-based, and hybrid architectures across most evaluation metrics. In particular, PCA-DRN achieves the lowest prediction errors and the strongest correlation consistency, indicating the complementary advantages of residual learning and meteorological feature compression.

Finally, the Bootstrap-based statistical significance analysis confirms that the observed performance differences are statistically reliable. The improvements obtained when increasing the mini-batch size from 32 to 64 are statistically significant for both DRN and PCA-DRN. Moreover, PCA-DRN maintains a statistically significant advantage over the original DRN under identical batch size settings. These results provide strong statistical support for the empirical findings reported in this section.

In summary, the experimental results demonstrate that mini-batch size plays a critical role in determining the forecasting performance of residual-network-based STLF models. Appropriate batch size selection can improve both optimization stability and generalization capability, while PCA-based meteorological feature compression provides additional performance gains. These findings establish a solid empirical foundation for the concluding discussion presented in the next section.

Conclusion

This study presented a systematic empirical investigation of mini-batch size sensitivity in DRNs for STLF. Using the DRN and PCA-DRN frameworks as representative residual-based forecasting models, the influence of mini-batch size on model training behavior and forecasting performance was examined under a unified experimental setting. Unlike most existing studies that primarily focus on architectural modifications or feature-level enhancements, this work explicitly treated mini-batch size as a key training configuration parameter and evaluated its practical impact on forecasting accuracy.

The experimental results demonstrate that mini-batch size exerts a substantial influence on the predictive performance of residual-based forecasting models. Across datasets representing both temperate (ISO-NE) and tropical (MyPJ) electricity systems, medium-scale mini-batch training consistently achieved a more favorable balance between optimization stability and generalization capability. In particular, a mini-batch size of 64 produced the best overall forecasting performance for both DRN and PCA-DRN, while a batch size of 32 provided a stable and competitive alternative. These findings indicate that appropriate batch size selection is an important factor for improving forecasting reliability in DRN-based STLF across different climatic environments.

Further analysis shows that PCA-DRN consistently outperforms the original DRN under identical training configurations. By incorporating multiple meteorological variables through PCA-based dimensionality reduction, PCA-DRN effectively reduces feature redundancy while preserving the residual learning structure of the original model. The resulting improvement in forecasting accuracy is statistically significant, confirming the practical benefit of meteorological feature compression in STLF applications.

Comparative experiments with several representative DL baseline models—including convolutional, recurrent, attention-based, and hybrid architectures—further highlight the effectiveness of residual learning frameworks. Under a unified training configuration, both DRN and PCA-DRN demonstrate superior or highly competitive forecasting performance across most evaluation metrics. Among all evaluated models, PCA-DRN achieves the lowest prediction errors and the highest fitting consistency, demonstrating the complementary advantages of residual learning and meteorological feature compression.

Bootstrap-based statistical significance testing further confirms the robustness of the empirical findings. The performance improvements observed when increasing the mini-batch size from 32 to 64 are statistically significant for both DRN and PCA-DRN. In addition, PCA-DRN consistently shows a statistically significant advantage over the original DRN under identical batch size configurations. These results indicate that the observed improvements are reliable and reproducible rather than incidental.

Despite these contributions, several limitations should be acknowledged. Although the experiments include datasets representing both temperate and tropical climatic conditions, the empirical evaluation remains limited to two real-world electricity systems. Future work may further validate the generality of the findings across additional power systems with different demand structures, climatic characteristics, and operational environments. In addition, to isolate the effect of mini-batch size, a fixed network architecture and optimization configuration were adopted throughout the experiments. Potential interactions between mini-batch size and other training hyperparameters—such as learning rate schedules, optimizer variants, or adaptive training strategies—were not systematically explored.

Future research may extend this work by examining mini-batch size sensitivity across broader benchmark datasets and diverse climatic regions. Moreover, investigating adaptive or dynamic mini-batch strategies may further improve training efficiency and model robustness in large-scale or real-time STLF applications.

In conclusion, this study demonstrates that mini-batch size is an important yet often overlooked training factor in residual-network-based STLF models. Proper batch size selection, together with effective meteorological feature compression, can lead to statistically robust improvements in forecasting accuracy. The findingQ1s provide practical guidance for training deep residual forecasting models and contribute to a clearer understanding of training configuration effects in DL–driven power system forecasting.

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. ISO-NE Dataset https://www.iso-ne.com/isoexpress/web/reports/. MyPJ Dataset Load: https://www.gso.org.my/SystemData/SystemDemand.aspx. Weather: https://www.met.gov.my/.

Received: 11 February 2026; Accepted: 16 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. Liu, J., Ahmad, F. A., Samsudin, K. & Hashim, F. Ab Kadir, M. Z. A. A comprehensive review of deep residual networks for short-term load forecasting. *Pertanika J. Sci. Technol.* **34**, 317–338 (2026).
3. 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).
4. 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).
5. 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).
6. 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).
7. 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).
8. 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).
9. 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).
10. 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).
11. 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 (2009). https://doi.org/10.1109/PES.2009.5275304
12. Zhang, Z., Zohren, S., Roberts, S. & DeepLOB Deep convolutional neural networks for limit order books. *IEEE Trans. Signal Process.* **67** (11), 3001–3012. https://doi.org/10.1109/TSP.2019.2907260 (2019).
13. Zaznov, I., Kunkel, J. M., Badii, A. & Dufour, A. The intraday dynamics predictor: A trioflow fusion of convolutional layers and gated recurrent units for high-frequency price movement forecasting. *Appl. Sci.* **14** (7), 2984. https://doi.org/10.3390/app14072984 (2024).
14. 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).

15. 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 (2017). https://doi.org/10.1109/ISPAN-FCST-ISCC.2017.78
16. 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).
17. Tang, X., Chen, H., Xiang, W., Yang, J. & Zou, M. Short-term load forecasting using channel and temporal attention based temporal convolutional network. *Electr. Power Syst. Res.* **205**, 107761. https://doi.org/10.1016/j.epsr.2021.107761 (2022).
18. Liu, M., Xia, C., Xia, Y., Deng, S. & Wang, Y. TDCN: A novel temporal depthwise convolutional network for short-term load forecasting. *Int. J. Electr. Power Energy Syst.* **165**, 110512. https://doi.org/10.1016/j.ijepes.2025.110512 (2025).
19. 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 (2017). https://doi.org/10.1109/SMC.2017.8123012
20. 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 (2018). https://doi.org/10.1109/IS.2018.8710498
21. 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).
22. 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).
23. 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).
24. 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 (2022). https://doi.org/10.1109/CIEEC54735.2022.9846658
25. 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).
26. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 770–778 (2016). https://doi.org/10.48550/arXiv.1512.03385
27. 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).
28. Kondaiah, V. Y. & Saravanan, B. Short-term load forecasting with deep learning. In Proc. Innov. Power Adv. Comput. Technol. (i-PACT), 1–5 (2021). https://doi.org/10.1109/i-PACT52855.2021.9696634
29. 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).
30. 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).
31. 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).
32. 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).
33. 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).
34. 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).
35. 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 (2020). https://doi.org/10.1049/oap-cired.2021.0257
36. 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).
37. 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).
38. Robbins, H. & Monro, S. A stochastic approximation method. *Ann. Math. Stat.* **22**, 400–407 (1951).
39. Duchi, J., Hazan, E. & Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.* **12**, 2121–2159 (2011).
40. Zaznov, I. et al. AdamZ: an enhanced optimisation method for neural network training. *Neural Comput. Applic.* **37**, 26887–26914. https://doi.org/10.1007/s00521-025-11649-w (2025).
41. Qian, X. & Klabjan, D. The impact of the mini-batch size on the variance of gradients in stochastic gradient descent. *arXiv* **arXiv:2004.13146** https://doi.org/10.48550/arXiv.2004.13146 (2020).
42. Masters, D. & Luschi, C. Revisiting small batch training for deep neural networks. *arXiv arXiv*. https://doi.org/10.48550/arXiv.1804.07612 (2018). :1804.07612.
43. Hwang, J. S., Lee, S. S., Gil, J. W. & Lee, C. K. Determination of optimal batch size of deep learning models with time series data. *Sustainability* **16**, 5936. https://doi.org/10.3390/su16145936 (2024).
44. Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M. & Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv arXiv*. 160904836. https://doi.org/10.48550/arXiv.1609.04836 (2016).
45. Marek, M., Lotfi, S., Somasundaram, A., Wilson, A. G. & Goldblum, M. Small batch size training for language models: When vanilla SGD works, and why gradient accumulation is wasteful. *arXiv arXiv*. 250707101. https://doi.org/10.48550/arXiv.2507.07101 (2025).
46. 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).
47. Huang, G. et al. Snapshot ensembles: Train 1, get m for free. arXiv 1704.00109 (2017). https://doi.org/10.48550/arXiv.1704.00109
48. 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).
49. Kingma, D. P., Ba, J. & Adam A method for stochastic optimization. *arXiv preprint arXiv*. https://doi.org/10.48550/arXiv.1412.6980 (2014). :1412.6980.
50. 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).

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/s41598-026-45002-5.

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

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.