

Article

# Research on a Non-Intrusive Load Recognition Algorithm Based on High-Frequency Signal Decomposition with Improved VI Trajectory and Background Color Coding <sup>†</sup>

Jiachuan Shi <sup>1,‡</sup>, Dingrui Zhi <sup>2,‡</sup>  and Rao Fu <sup>2,\*</sup> 

<sup>1</sup> Shandong Key Laboratory of Intelligent Buildings Technology, Shandong Jianzhu University, Jinan 250101, China; jc\_shi@sdjzu.edu.cn

<sup>2</sup> School of Information and Electrical Engineering, Shandong Jianzhu University, Jinan 250101, China; 2021080123@stu.sdjzu.edu.cn

\* Correspondence: furao20@sdjzu.edu.cn

<sup>†</sup> This paper is an extended version of our paper published in the 4th International Conference on Neural Computing for Advanced Applications, Hefei, China, 7–9 July 2023.

<sup>‡</sup> These authors contributed equally to this work.

**Abstract:** Against the backdrop of the current Chinese national carbon peak and carbon neutrality policies, higher requirements have been put forward for the construction and upgrading of smart grids. Non-intrusive Load Monitoring (NILM) technology is a key technology for advanced measurement systems at the end of the power grid. This technology obtains detailed power information about the load without the need for traditional hardware deployment. The key step to solve this problem is load decomposition and identification. This study first utilized the Long Short-Term Memory Denoising Autoencoder (LSTM-DAE) to decompose the mixed current signal of a household busbar and obtain the current signals of the multiple independent loads that constituted the mixed current. Then, the obtained independent current signals were combined with the voltage signals to generate multicycle colored Voltage–Current (VI) trajectories, which were color-coded according to the background. These color-coded VI trajectories formed a feature library. When the Convolutional Neural Network (CNN) was used for load recognition, in light of the influence of the hyperparameters on the recognition results, the Bayesian Optimization Algorithm (BOA) was used for optimization, and the optimized CNN network was employed for VI trajectory recognition. Finally, the proposed method was validated using the PLAID dataset. The experimental results show that the proposed method exhibited better performance in load decomposition and identification than current methods.

**Keywords:** denoising autoencoder; Bayesian optimization; non-intrusive load recognition; convolutional neural network

**MSC:** 86T20



**Citation:** Shi, J.; Zhi, D.; Fu, R. Research on a Non-Intrusive Load Recognition Algorithm Based on High-Frequency Signal Decomposition with Improved VI Trajectory and Background Color Coding. *Mathematics* **2024**, *12*, 30. <https://doi.org/10.3390/math12010030>

Academic Editor: Ke-Lin Du

Received: 15 November 2023

Revised: 18 December 2023

Accepted: 19 December 2023

Published: 22 December 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Given the ongoing increase in domestic electricity consumption, incorporating effective daily electricity usage planning is important to minimize energy waste. In this regard, NILM can help identify domestic electrical appliances and their states, enabling domestic consumers to have a better understanding of their electricity consumption behavior and improve their usage patterns [1]. Moreover, NILM can also enhance the management and optimization of the demand side of the power grid to achieve energy saving goals. Therefore, NILM holds great significance for energy conservation and emission reduction [2]. Unlike traditional methods requiring the installation of multiple sensors, NILM integrates electricity information into a single collection signal, making it more cost-effective and practical [3]. NILM can be primarily divided into two categories based on its objectives.

The first is load identification, which involves recognizing the electrical appliances that users are currently using or identifying changes in the switch status of the appliances. The second is load decomposition, where the total power is broken down into the power consumption of individual electrical devices. By analyzing information from these two categories, NILM can provide users and power suppliers with accurate device-level electricity consumption details, including startup and shutdown times, operating duration, power consumption, and the associated costs. In terms of event changes, NILM can be classified into event-based load identification and non-event-based load decomposition [4]. Event-based load identification methods typically include three steps: event detection, feature extraction, and load classification [5]. Non-event-based load decomposition usually does not require event detection; instead, it directly uses the total power sequence as the model input. This is because its purpose is not to identify the electrical appliances running during a specific time period but to decompose the total power based on the differences in the power and operating modes of different appliances, directly obtaining the power sequences of individual electrical devices.

Regarding load identification technology, a 2019 study [6] utilized the Dynamic Time Warping (DTW) algorithm to calculate the similarity between test templates and reference templates, using transient waveforms and the power change values of household load switching events as feature quantities for load recognition. A more recent study [7] clustered load data based on steady-state values and proposed a non-intrusive load recognition method using Feature Weighted K-Nearest Neighbor (FWKNN), which improved the feature distance calculations with feature weights and achieved high accuracy. A further study [8] presented an attention model that combined global and sliding window approaches, employing bidirectional Long Short-Term Memory (LSTM) networks as encoders to extract information, utilizing an attention mechanism to capture the current load information and decode the output decomposition results. A 2020 study [9] encoded different power states of target appliances using a deep recurrent convolutional model, extracting spatiotemporal features of input total load power, and implemented state modeling of different target appliances through transfer learning, achieving significant improvements compared to Markov models. In another recent study [10], a lightweight recognition task was accomplished by combining CNN and K-Nearest Neighbor (KNN) networks. Simultaneously, for unknown loads, user feedback may not be required, allowing the system to operate in unfamiliar home environments. Even more recently [11], a novel method was proposed to extract distorted images of current harmonics from aggregated signals over 60 cycles, enabling appliance classification within 1 s with low computational complexity. Earlier, a study [12] described in detail eight shape features such as the asymmetry observed in VI trajectories and conducted recognition using hierarchical clustering. Subsequently, researchers [13] used the elliptical Fourier descriptor of VI trajectory contours as input for a classification algorithm. A further study [14], based on [12], introduced a new feature called “span” and compared it with the harmonic content, active power, and reactive power of the current, utilizing four classification algorithms. The results showed that the VI trajectories had good recognition capability. The authors of [15] introduced a transfer learning methodology grounded in the VI trajectory to address the challenge of limited data label acquisition. Additionally, the VI trajectory underwent a transformation into a visual representation through color encoding. This process not only heightened the uniqueness of the appliance signature but also facilitated the integration of transfer learning into NILM. The authors of [16] proposed adaptive weighted recurrence graph blocks for the representation of appliance signatures in event-based NILM. By converting the activation current of a single cycle into a weighted recursive graph, the proposed method ensured the distinctiveness of appliance signatures. Consequently, this approach demonstrated superior recognition performance on the LILACD and PLAID public datasets compared to traditional VI-trajectory-based methods. A reconstructed VI trajectory was introduced, incorporating the Particle Swarm Optimization (PSO) algorithm to ascertain the optimal threshold parameters. This enhancement aims to maximize the model’s classification

capability, addressing the challenge faced by conventional VI-trajectory-based methods in effectively distinguishing similar appliances [17]. A 2021 study [18] introduced the Asymmetric Deep Supervised Hashing (ADSH) method, utilizing VI trajectory signatures for NILM. This approach employed the VI trajectory as the model input, effectively addressing challenges related to the inefficient computation of massive data and the limited discrimination of manually extracted signatures. Simultaneously, an asymmetric learning architecture was applied for hash code learning. Specifically, a convolutional neural network model was used for high-dimensional feature extraction and hash function learning in some training trajectories to establish coding rules. For all training trajectories, the direct learning of coding rules ensured consistency in the codes, thereby achieving accurate appliance recognition. This approach significantly enhanced recognition accuracy while maintaining a small code length and reducing the space complexity. To tackle the difficulties associated with acquiring label data and correctly identifying the VI trajectory of multistate appliances, a semi-supervised learning method based on the semi-supervised teacher graph network [19] was proposed. This method compacted the feature distribution of multistate appliances by constructing a teacher graph, leading to improved recognition results. However, while significant progress has been made in the accuracy of load identification using VI trajectories as features, several challenges persist. Firstly, the current approach normalizes the voltage and current data to derive the VI trajectories, leading to a loss of energy information. Secondly, appliances with continuous power variations, such as computers, are difficult to represent using steady-state VI trajectories. Lastly, VI trajectories fail to capture the stability characteristics of a load's operational state during runtime, specifically the amplitude of the current oscillations.

In the realm of load decomposition techniques, traditional methods rely on low-frequency data to decompose aggregated data into device-level information. This process does not necessitate event detection and typically focuses on power data as the primary research object. The Hidden Markov Model (HMM) has been a commonly used regression model. Kim et al. validated four HMM extension models and introduced non-electrical characteristics such as the load switching duration, load usage frequency, and interdependence among loads, which improved the decomposition accuracy to a certain extent [20]. Kolter et al. applied the Factorial Hidden Markov Model (FHMM) algorithm to establish a load decomposition model, using the total current signals as the research object. When a signal change was detected, the signal differences between the load currents were encoded, resulting in the optimal solution for the model and achieving a high decomposition accuracy [21]. With the continuous development of deep learning technology, non-intrusive load decomposition models based on deep learning have emerged [22]. In 2015, Kelly introduced deep learning methods into the NILM field and coined the term "Neural NILM". Kelly proposed three neural network structures: LSTM, Recurrent Neural Network (RNN), and Distributed Agent Environment (DAE), which were effective in handling long time-series data [23]. Roberto improved Kelly's DAE model by combining it with a median filter, which partially eliminated the noise interference and improved the model's robustness [24]. Odysseas modified Kelly's LSTM network to tackle the issue of the high number of network neurons, which led to long training times and thus, its unsuitability for embedded devices. Instead, the Gated Recurrent Unit (GRU) algorithm was used, reducing the network's depth while maintaining the accuracy and reducing the computational complexity [25]. Zhang proposed a load decomposition framework that mapped sequences to individual load points, with aggregated power data of windows as the network's input and the corresponding single-point load data as the network's output. This mapping pattern accelerated the training speed and improved the load decomposition's real-time performance [26]. Barsim proposed a general deep decomposition model that automatically adjusted the model parameters based on the characteristics of different loads, thus obtaining more accurate load-switching state sequences [27]. Piccialli adopted a neural network model with an attention mechanism, which strengthened the correlation between the input and output and improved the decomposition accuracy [28].

The tasks of load decomposition and load identification have conventionally been regarded as independent processes. However, during load identification tasks, it is imperative to conduct load decomposition. Distinctive load characteristics can only be further extracted by obtaining the independent electrical signals of individual loads through load decomposition. In the load identification process, the commonly employed method for decomposition is the differencing technique. This involves taking the difference between the total voltage and current data before and after the occurrence of an “event”. However, practical challenges arise in obtaining accurate current–voltage data through differencing during actual operations, particularly when loads with substantial fluctuations exist on the user bus. Obtaining precise current signals is especially challenging. Consequently, this study introduces the notion of traditional load decomposition into the load identification task to replace the differencing method. Given that voltage signals are minimally affected by load switch actions, only current signals require decomposition. Hence, a deep learning approach is employed for current decomposition, obtaining independent current signals. The current decomposition in load identification differs from traditional load decomposition, which employs low-frequency signals with a sampling frequency typically ranging from 1/6 Hz to 1 Hz. Traditional methods require neural networks to learn the features of the entire load operating cycle over an extended period. In contrast, load identification necessitates high-frequency features, often employing data exceeding 1 kHz. As high-frequency signals exhibit strong periodicity, this study transforms the decomposition issue in load identification into a denoising problem. By denoising mixed current signals, independent current signals for individual loads are obtained.

Thanks to the higher discriminability of electrical quantities in VI trajectories compared to other features and its effective integration with image recognition, the complexity of feature extraction and load identification was reduced. Consequently, the current study opted for VI trajectories as the distinctive feature for load recognition. Despite the significant progress achieved in load recognition accuracy by using VI trajectories as features, the existing issues still exerted a considerable impact on the precision of load identification.

To address these issues, this paper proposed an approach based on LSTM-DAE. This method was employed to decompose complex mixed current signals and obtain independent current signals of unknown loads. The decomposed current signals were combined with voltage signals, and the resulting VI trajectories were plotted as features for load identification. Recognizing the limitations inherent in VI trajectories, traditional single-cycle VI trajectories were enhanced by transforming them into multicycle VI trajectories to capture fluctuations during load operation. Additionally, color coding was applied to VI trajectories to enhance the distinctiveness between different loads. Different background colors were assigned to VI trajectories based on varying current amplitudes, addressing the issue of energy information loss caused by the normalization of current–voltage data.

Following the completion of VI trajectory feature extraction, the improved VI trajectories were subjected to image recognition using the AlexNet architecture within a CNN to accomplish the load identification task. Given the numerous parameters of the AlexNet network and their significant influence on recognition accuracy, the BOA was employed to optimize the hyperparameters of the AlexNet network, aiming to achieve the optimal solution. The specific methodology was as follows:

- (1) The high-frequency mixed current signals were decomposed using the LSTM-DAE. This work accurately acquired the current signals of each load by exploiting the high sensitivity of LSTM to temporal signal features and the ability of the DAE to transform the load decomposition problems into denoising problems, which were then utilized as the fundamental data for load recognition;
- (2) Colored VI trajectories were generated by plotting the VI trajectories obtained from the multicycle voltage and current data. The R channel represented the normal multicycle VI trajectory, the G channel represented the current variation slope between adjacent sampling points, and the B channel represented the rate of power changes.

Additionally, the VI trajectory background was color processed based on the difference in the current amplitude to obtain a multicycle color-encoded VI trajectory feature library with filled background colors;

- (3) The VI trajectory feature library was transformed into an  $n \times n$  image format and input into the AlexNet network for training. Since the traditional AlexNet network was not suitable for load recognition tasks, the BOA algorithm was employed to optimize the network parameters, thereby achieving better recognition performance;
- (4) The PLAID dataset was utilized in the experiments, and the results demonstrated that the six selected load decomposition accuracies all exceeded 94.

## 2. LSTM-DAE-Based Load Decomposition

### 2.1. Load Decomposition

Load decomposition is the process of decomposing one mixed telecommunications signal into the independent telecommunications signals for each load. This research transformed this process into a denoising problem to obtain accurate information about the current of each load. Assuming there were  $N$  loads, the total current [29] at time  $t$  was:

$$I(t) = \sum_{k=1}^N I_k(t) + I_\delta(t), \tag{1}$$

where  $I(t)$  was the total current at the current moment,  $I_k(t)$  was the current generated by the  $k$ th load, and  $I_\delta(t)$  was the noise disturbance.

Assuming the target load current represented the actual signal and considering the total current signal as the composite of the actual signal and noise interference, the equation was modified as follows:

$$I(t) = I_\eta(t) + I_\beta(t), \tag{2}$$

where  $I_\eta(t)$  is the current of the target load and also the real current in the denoising process, and  $I_\beta(t)$  is the noise in the denoising process, which is the superposition of the currents of the loads other than the current of the target load and the real noise, expressed as:

$$I_\beta(t) = \sum_{k=1}^{N-1} I_k(t) + I_\delta(t). \tag{3}$$

### 2.2. Denoising Autoencoder

As shown in Figure 1, the Autoencoder (AE) had an equal number of nodes for the input and output, with the aim of minimizing the reconstruction error between the input and output. The AE consisted of an encoder  $\phi$  and a decoder  $\psi$ , where the encoder mapped the input to a low-dimensional feature space, and the decoder mapped the data from the low-dimensional feature space back to the original input. In Figure 1,  $x$  represented the input data to the encoder,  $h$  represents the data mapped from  $x$  to the hidden feature space, and  $x'$  represents the data reconstructed by the decoder using the hidden feature space data  $h$ . Assuming the input space was  $I = R^n$  (where  $n$  was the number of variables), the working principle of the autoencoder was defined as follows:

$$\begin{aligned} \phi &: I \rightarrow F \\ \psi &: F \rightarrow I \\ \phi, \psi &= \underset{\phi, \psi}{\operatorname{argmin}} \|x - x'\|^2 \end{aligned} \tag{4}$$

where  $\|\cdot\|$  denoted the Frobenius norm. Taking an AE with one hidden layer as an example, the encoder mapped the input  $x \in I$  to the hidden feature space  $h \in F = R^p$  ( $p$  was the dimension of the hidden layer), and the mapping process was

$$h = \sigma(Wx + b), \tag{5}$$

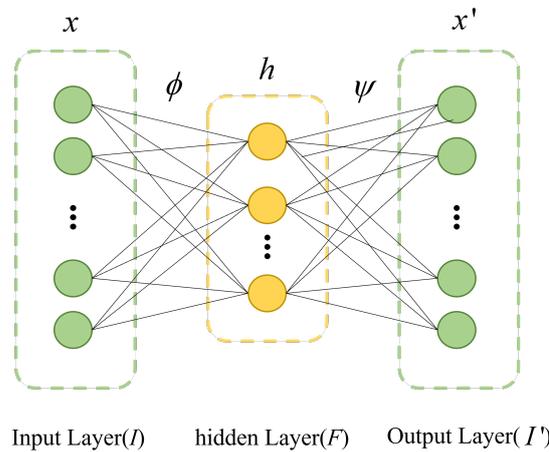
where  $\sigma$  was the activation function in the encoder,  $\mathbf{W}$  was the weight matrix, and  $\mathbf{b}$  was the bias vector. After the encoder mapped the input variable to the hidden layer, the decoder reconstructed the variable  $x'$  into a variable of the same size as the input variable  $x$  based on the information in the hidden layer  $h$ . The mapping process was as follows:

$$x' = \sigma'(W'x + b'), \tag{6}$$

where  $\sigma'$  represented the activation function in the decoder, and  $\mathbf{W}'$  and  $\mathbf{b}'$  corresponded to the weight matrix and deviation vector in the decoder, respectively. The goal of the self-encoder was to find the process that minimized the residuals between the input and the reconstructed quantity  $x'$ :

$$\begin{aligned} L(x, x') &= \frac{1}{m} \|x - x'\|^2 \\ &= \frac{1}{m} \|x - \sigma'(W'\sigma(Wx + b)) + b'\|^2, \end{aligned} \tag{7}$$

where  $m$  represented the number of sample points in the AE network. The optimization problem of the self-encoder was solved by backpropagation.



**Figure 1.** Basic structure of an Autoencoder.

The Denoising Autoencoder (DAE) is a variant of an AE first proposed by Vincent et al. [30]; its network structure is shown in Figure 2. The difference between the DAE and an AE is that it uses partially corrupted data to train the DAE network in order to recover the real inputs, and its function is to separate and remove the noise in the data to obtain the real data. The workflow of a DAE is as follows:

- (1) We obtain the corrupted data by adding noise to the normal data or randomly discarding parts of the normal data;

$$\tilde{x} = q(\tilde{x} | x) \tag{8}$$

- (2) In this study, the corrupted data  $\tilde{x}$  is mapped to the low-dimensional hidden feature space by the self-encoder coding process  $h = \sigma(Wx + b)$ ;
- (3) The decoder decodes the corrupted data's mapping in the hidden feature space using Equation (6) and obtains the reconstructed data;
- (4) The minimization problem of Equation (7) is solved using the backpropagation algorithm.

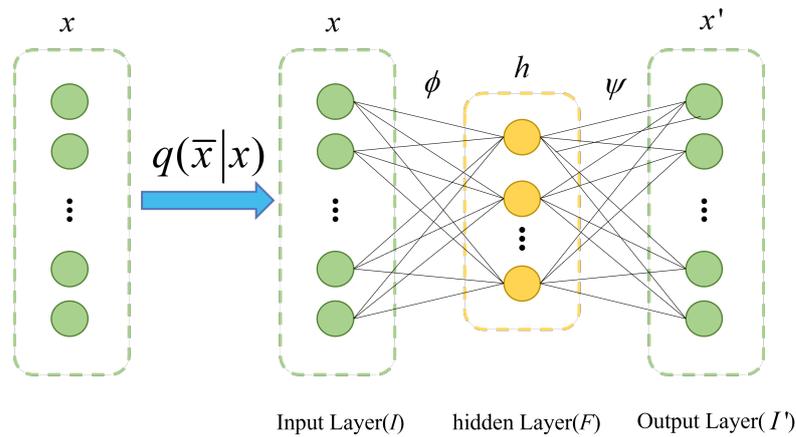


Figure 2. Structure of the Denoising Autoencoder.

In this way, in this paper, signal decomposition was converted into a denoising problem. Specifically, the mixed current signals were treated as noisy signals, and the independent load signals were seen as real data after undergoing denoising processing.

2.3. Decomposition Model Based on LSTM-DAE

The LSTM network has exhibited good performance in extracting features from time-series signals [31,32]. Therefore, LSTM was combined with DAE. The encoder and decoder parts of the DAE were composed of a dual-layer LSTM and Dropout layers, respectively. The input data were the mixed current data, and a sliding window approach was employed in which multiple cycles of a mixed current were used as inputs. This work selected a window size of 10 cycles to achieve the functionality of seq2seq and obtain the output of the independent current. The structure is shown in Figure 3.

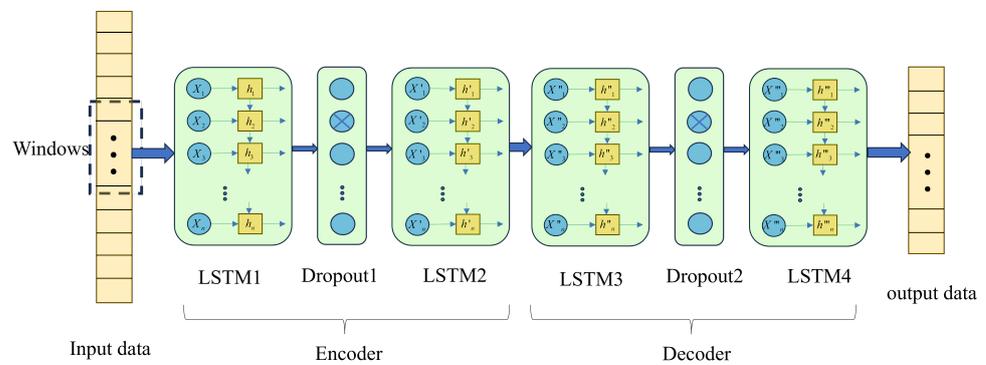


Figure 3. The LSTM-DAE neural network model.

3. Load Recognition of VI Traces Based on Background Color Coding

3.1. Construction of the VI Trajectory Pixelization

The VI trajectory refers to the voltage–current relationship curve used to describe the performance characteristics of electrical equipment. Traditional VI trajectory analysis methods mapped the curve onto a unit grid, where each grid cell represented the presence or absence of VI trajectory information (0 or 1). The grid was then transformed into an image, thereby converting the load recognition problem into an image recognition problem for automated identification. However, previous methods only utilized binary values to represent each grid cell, resulting in relatively limited information content. Therefore, the current study proposes an improved approach aimed at enhancing the expressive power of the VI trajectory data.

- (1) Conventional VI traces typically analyzed changes in a single cycle, which failed to capture the characteristic changes of the load across different operating cycles.

Therefore, this work utilized 20 consecutive cycles of current–voltage signals to generate VI traces, aiming to provide a more comprehensive reflection of the changes across different cycles;

- (2) Color coding the VI trajectory: The multicycle VI trajectory was represented in red (R channel), the slope of the straight line segment between adjacent sampling points of the VI trajectory was represented in green (G channel), and the instantaneous power value was represented in blue (B channel), thereby generating a VI trajectory image with colored tracks;
- (3) Due to the significant differences in current amplitude between certain loads with similar VI trajectories, and considering that current amplitude is an important load characteristic, this research assigned different colors to the background of the VI trajectories based on the varying current amplitudes in order to highlight the differences in current amplitude.

The specific steps were as follows:

- (1) In this study, the voltage and current values were standardized, and the resulting standardized data were used to plot the standardized VI trajectory. The standardization formula employed in this research is as follows:

$$\Delta V_m = \frac{V_m}{\max|V|} \tag{9}$$

$$\Delta I_m = \frac{I_m}{\max|I|}, \tag{10}$$

where  $\max|V|$  was the maximum value of the voltage in the steady-state sequence and  $\max|I|$  was the maximum value of current in the steady state sequence.  $V_m$  and  $I_m$  were the voltage and current values of the  $m$ th sampling point in the sequence, respectively, and  $\Delta V_m$  and  $\Delta I_m$  were the voltage and current values of the  $m$ th sampling point after normalization, respectively;

- (2) In this study, the VI trajectory was created using normalized data, and the resultant VI trajectory served as the R channel of the colored VI trajectory. Subsequently, the green (G) and blue (B) channels were established in sequence, facilitating the amalgamation of the RGB channels to form a colorful VI trajectory;
- (3) In this study, the G channel was created by mapping the slope of the straight line segments to the (0, 1) range using the arctan function.

$$K_j = \frac{i_f(j+1) - i_f(j)}{V(j+1) - V(j)} \times \frac{\max(|V|)}{\max(|I_f|)} \tag{11}$$

$$G_j = \frac{1}{2} + \frac{\arctan(K_j)}{\pi}, \tag{12}$$

where  $K_j$  was the slope of the  $j$ th straight line segment, and  $G_j$  was the G-channel depth value of the  $j$ th straight line segment.  $i_f(j)$  signified the current value at the present data point, and  $i_f(j+1)$  represented the subsequent data point’s current value.  $V_f(j)$  corresponded to the voltage value at the current data point, while  $V_f(j+1)$  represented the voltage value at the next data point.  $\max|V|$  was utilized to extract the absolute value of the maximum voltage obtained from the data used to plot the VI trajectory, and  $\max|I_f|$  was employed to extract the absolute value of the maximum current obtained from the data depicting the VI trajectory.

The  $G_j$  was mapped to the VI trajectory to obtain the corresponding G-channel depth value  $G_{m,n}$  for each grid, which was then normalized to obtain the G-channel value for each grid point;

$$G_{m,n}' = \frac{G_{m,n}}{\max G} \tag{13}$$

$$\max G = \max\{G_{1,1}, G_{1,2}, G_{1,3}, \dots, G_{n,n}\} \tag{14}$$

- (4) The B channel was created with the following instantaneous power values:

$$P_j = I_j V_j \tag{15}$$

Mapping  $P_j$  onto the VI trajectory results in depth values  $P_{m,n}$  for each grid. Following this, the multiperiod power was normalized.

$$P_{m,n}' = \frac{P_{m,n}}{\max P} \tag{16}$$

$$\max P = \max\{P_{1,1}, P_{1,2}, P_{1,3}, \dots, P_{n,n}\}, \tag{17}$$

where  $P_j$  was the instantaneous power value,  $P_{m,n}$  was the power superimposed on each grid in the grid,  $\max P$  was the maximum value of power in all the grids, and the resulting  $P'_{m,n}$  was the value of the B channel;

- (5) For the addition of the background color, the average of the RMS values of the current energy of 25 adjacent cycles was obtained and matched with the set background color to determine the background color.

Figure 4 shows the multicycle monochromatic VI trajectory, the multicycle color-coded VI trajectory, and the color-coded VI trajectory with background fill.

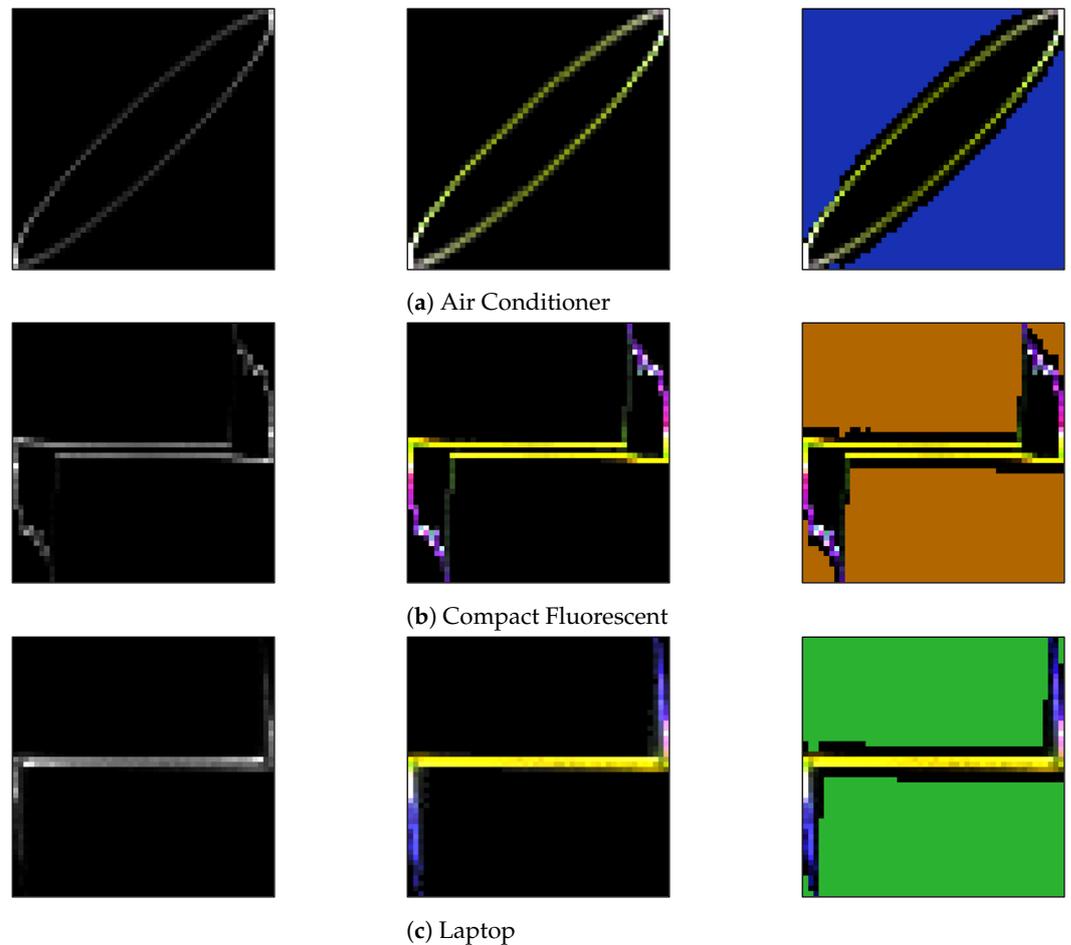
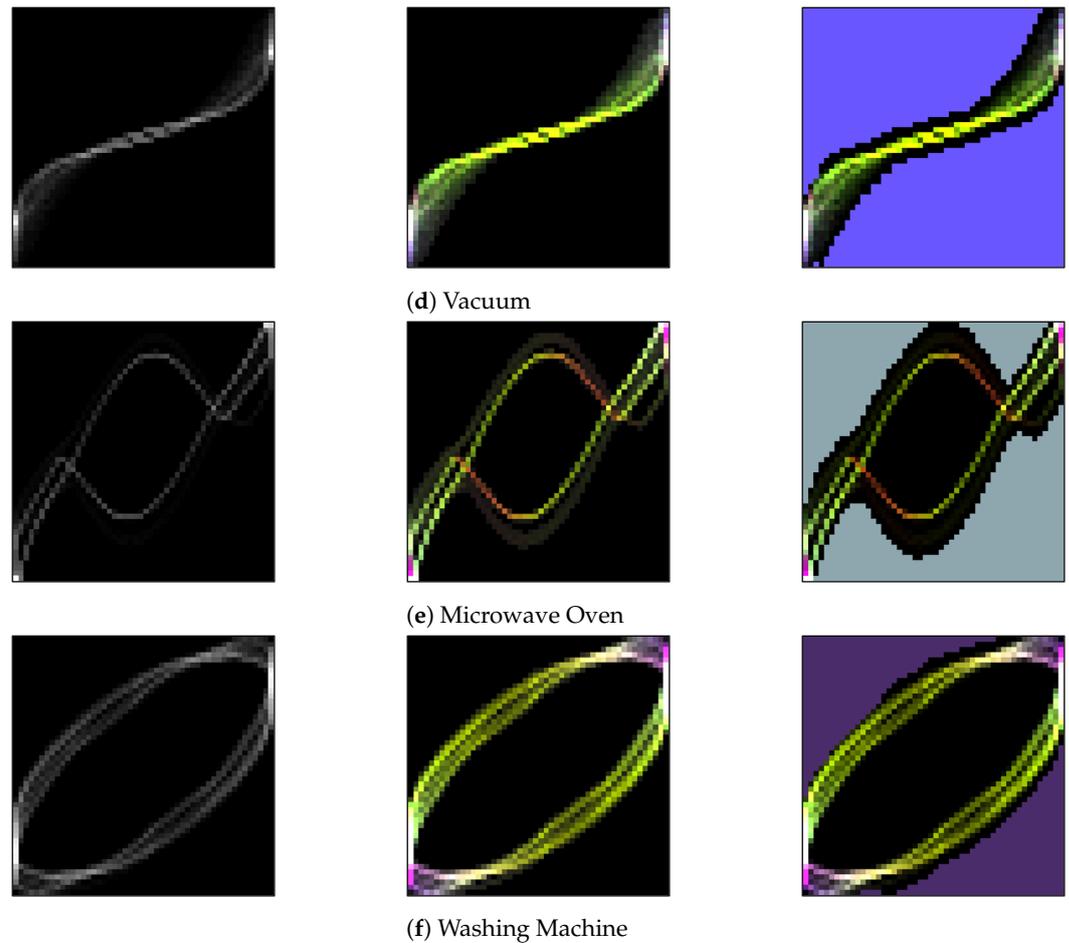


Figure 4. Cont.

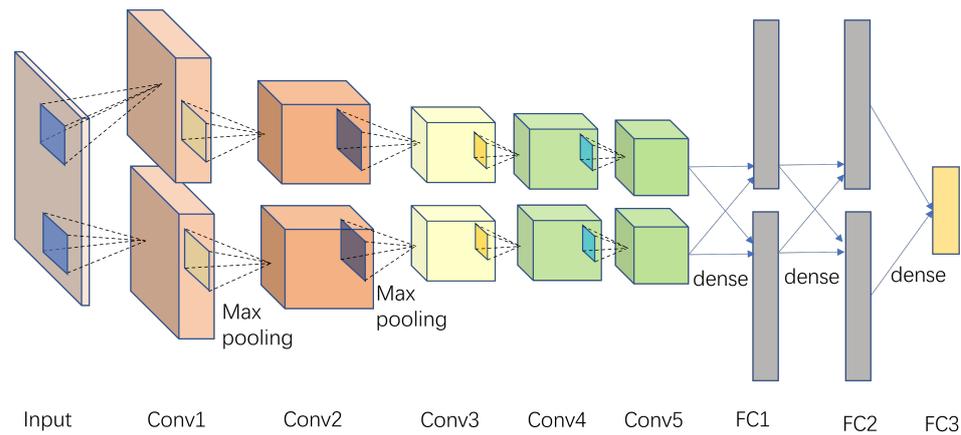


**Figure 4.** Monochromatic VI trajectory, VI trajectory with color coding, and color-coded VI trajectory with background colors.

### 3.2. Construction of a Convolutional Neural Network

CNN possess outstanding advantages in handling two-dimensional input data [33,34]. Therefore, the AlexNet network, proposed in 2012, was chosen for load recognition, as it represented a significant breakthrough in deep learning in the field of image recognition. The structure of this model consisted of five convolutional layers, three pooling layers, two Dropout layers, and three fully connected layers, with the activation function being Rectified Linear Units (ReLU). To accomplish the recognition task, the encoded VI trajectory image with filled background color was used as the model input (an  $n \times n$  matrix, where  $n = 50$ ). The schematic diagram of the AlexNet network is shown in Figure 5.

The classic AlexNet network could not be directly applied to load recognition tasks and required some modifications to make it work properly. Additionally, due to the numerous hyperparameters, there was a need to optimize the hyperparameters of the convolutional layers, pooling layers, and fully connected layers in order to obtain a highly accurate recognition model.



**Figure 5.** AlexNet network model diagram.

### 3.3. Bayesian Optimization Algorithm

Bayesian Optimization (BO) is used to estimate the maximum value of a function based on existing sampled points when the functional equation is unknown [35]. It effectively addresses the classical problem of finding the next evaluation point based on the information acquired about the unknown objective function, to quickly search for the optimal solution [36]. Bayesian Optimization has been highly applicable in evaluating costly and complex optimization problems and has been widely used in the optimization of machine learning hyperparameters, deep learning model hyperparameters, and other related areas. In this study, the hyperparameters selected for optimization were the number of convolutional kernels, the size of the convolutional kernels, the stride of the convolutional kernels, the size of pooling kernels, the stride, and the Dropout probability. The optimization ranges are presented in Table 1. The parameter definitions for Bayesian Optimization are provided in Table 2.

**Table 1.** Hyperparameters to be optimized and their ranges.

Layers	Hyperparameters	Dynamic Range
Conv	Number of convolution kernels	30~135
	Convolution kernel size	2~6
	Convolution kernel step	1~3
Pool	Pool core size	2~6
	Pool nucleation step size	1~3
Dropout	Dropout rate	0~1

**Table 2.** Parameter definitions in the Bayesian Optimization algorithms.

Layers	Hyperparameters
a1, a2, a3, a4, a5	The number of convolution kernels in five convolutional layers
b1, b2, b3, b4, b5	Convolutional kernel size for five convolutional layers
c1, c2, c3, c4, c5	Convolutional kernel step size for five convolutional layers
d1, d2, d3	The number of pooling kernels in three pooling layers
e1, e2, e3	Step size of pooling kernels in three pooling layers
f1, f2	Dropout rate of the two layers

#### 4. Dataset and Evaluation Criteria

This research validated the effectiveness of the algorithm using the PLAID dataset from Carnegie Mellon University in the United States. The dataset included instantaneous values of current and voltage for 11 different types of appliances, sampled at 30 kHz, recorded in multiple households in Pittsburgh, Pennsylvania. In this dataset, mixed currents with multiple loads are missing, and only individual current data when each load operated separately are available. Therefore, before analysis, it was necessary to align and superimpose the current values based on the voltage values to obtain mixed currents. This article selected six loads for load decomposition verification, namely, an air conditioner, an energy-saving lamp, a laptop, a vacuum cleaner, a microwave oven, and a washing machine.

#### 5. Experimental Analysis

##### 5.1. Assessment of Indicators

##### 5.1.1. Evaluation Indexes of Decomposition Process

The root mean square error (RMSE), mean absolute error (MAE), phase error, and correlation coefficient were selected as the evaluation metrics. The RMSE is a commonly used metric that measures the mean square difference between the predicted and actual values, representing the average magnitude of the prediction errors. A lower RMSE value indicates that the predicted results are closer to the actual values. The MAE is another common evaluation metric that measures the average absolute difference between the predicted and actual values, reflecting the absolute error of the predictions. The phase error is an important metric for high-frequency current decomposition, measuring the error between the predicted and actual phase of the signals. The correlation coefficient is used to measure the linear correlation between the predicted and actual values. A correlation coefficient closer to 1 indicates a better linear relationship between the predictions and the actual values. The definitions of these four evaluation metrics are as follows:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (I_i - \hat{I}_i)^2}; \quad (18)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |I_i - \hat{I}_i|; \quad (19)$$

$$Phase\ Error = \arccos \left( \frac{\sum_{i=1}^n I_i \hat{I}_i}{\sqrt{\sum_{i=1}^n I_i^2 \sum_{i=1}^n \hat{I}_i^2}} \right); \quad (20)$$

$$Correlation = \frac{\sum_{i=1}^n (I_i - \bar{I})(\hat{I}_i - \bar{\hat{I}})}{\sqrt{\sum_{i=1}^n (I_i - \bar{I})^2 \sum_{i=1}^n (\hat{I}_i - \bar{\hat{I}})^2}}. \quad (21)$$

##### 5.1.2. Evaluation Metrics for the Load Recognition Process

For the evaluation of non-intrusive load identification performance, common evaluation metrics such as recognition accuracy ( $Acc$ ), F1 score, and  $F$ -measure [37] were primarily utilized. The calculation formulas are depicted in Equations (22)–(25).

$$Acc = \frac{m}{n}, \quad (22)$$

where  $m$  denotes the number of correct classifications of the model, and  $n$  is the total number of samples;

$$F_{1_i} = 2 \cdot \frac{precision \cdot recall}{precision + recall}; \quad (23)$$

$$precision = \frac{TP}{TP + FP}; \quad (24)$$

$$recall = \frac{TP}{TP + FN}. \quad (25)$$

In the above equation,  $TP$  represents the number of true positives, where both the true value and the predicted value are positive;  $FP$  represents the number of false positives, where the true value is negative, but the predicted value is positive; and  $FN$  represents the number of false negatives, where the true value is positive, but the predicted value is negative. The following equation was used to obtain the average value for each device:

$$F_{i,mean} = \frac{1}{L} \sum_{g=1}^L F_{1,g,i}. \quad (26)$$

In this equation,  $L$  represents the total number of occurrences of device  $i$  in the test set;  $F_{1,g,i}$  represents the  $F_1$  value of device  $i$  in the  $g$ -th occurrence; and  $F_{i,mean}$  represents the average  $F_1$  score for the  $i$ -th device over  $L$  trials, explicitly portraying the F-measures for device  $i$ . Finally, by calculating the average of all the F-measures using Equation (26), the macro-average value  $F_{macro}$  is obtained.

$$F_{macro} = \frac{1}{A} \sum_{i=1}^A F_{i,mean}, \quad (27)$$

where  $A$  represents the total number of different equipment types.

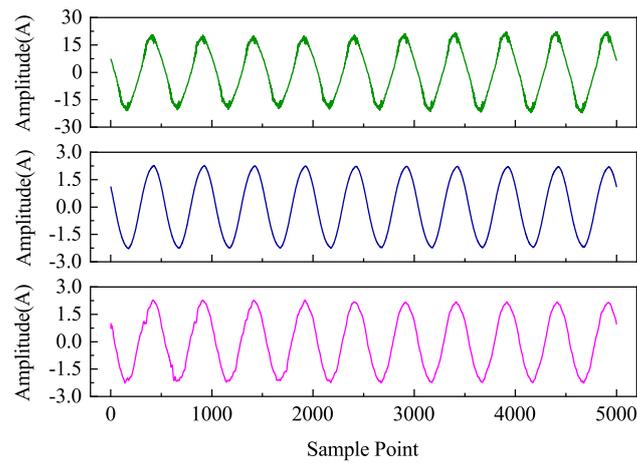
## 5.2. Example Analysis

### 5.2.1. Load Decomposition

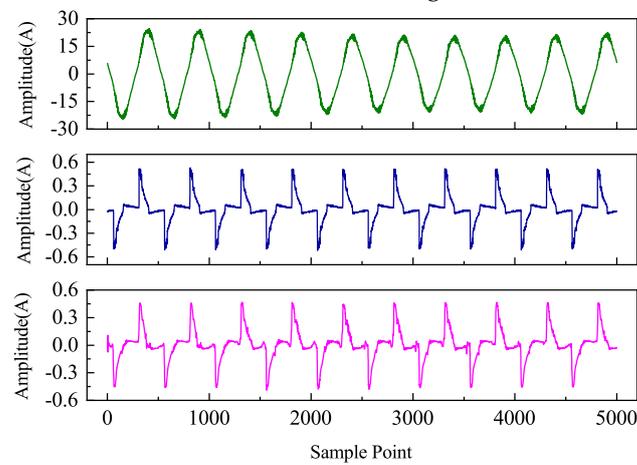
The input of the LSTM-DAE model was a mixed current obtained by superimposing multiple load currents. The window size of the data input was set to 10 cycles of current data. In the PLAID dataset, one cycle contains 500 data points. Therefore, the size of each window was set to 5000. The sliding step was one cycle of data, which was 500 data points. The training times (epochs) were set to 1000. The decomposition results are shown in Figure 6. Each figure contains three curves representing, from top to bottom, the mixed current, the true target load current, and the predicted target load current.

From Figure 6, it can be observed that the latent information of strongly correlated current sequences was effectively extracted, and the temporal features within the time-series signals were mined using the LSTM-DAE network model. Even in cases where irregular variations were present in the mixed signals and actual signals, the target signal was accurately decomposed by the model. Overall, the decomposed current signals output by this model were well able to track the variations in the actual current signals and essentially fit the rising and falling trends of the actual current curves. These results suggest that the decomposition and fitting of current signals were performed effectively by the model.

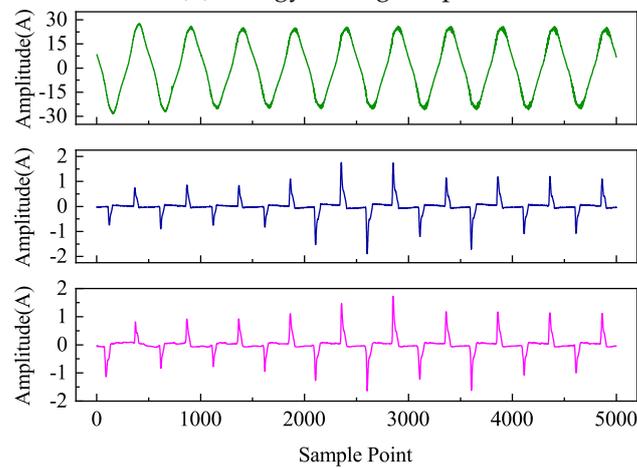
From Table 3, it can be seen that out of the six types of loads, five had correlation coefficients exceeding 98%. Even the load with the lowest correlation coefficient, which belonged to the laptop (notebook) category, had a coefficient close to 95%. This indicated a strong linear relationship between the predicted results of the algorithm and the actual values, effectively capturing the changing trends in the load current. Additionally, the maximum values for the RMSE, MAE, and phase error were 0.866, 0.551, and 0.332, respectively. In comparison to the four network models listed in Reference [38], the algorithm in this study achieved a minimum MAE score of 5.97 and a minimum RMSE of 10.52, demonstrating a significant improvement in algorithm performance. Overall, the algorithm exhibited high precision, a low phase error, and a strong correlation in current signal decomposition tasks, making it particularly suitable for handling high-frequency currents. These advantages give the algorithm potentially high practical value in applications such as power load analysis and energy management, providing a reliable tool.



(a) Air conditioning

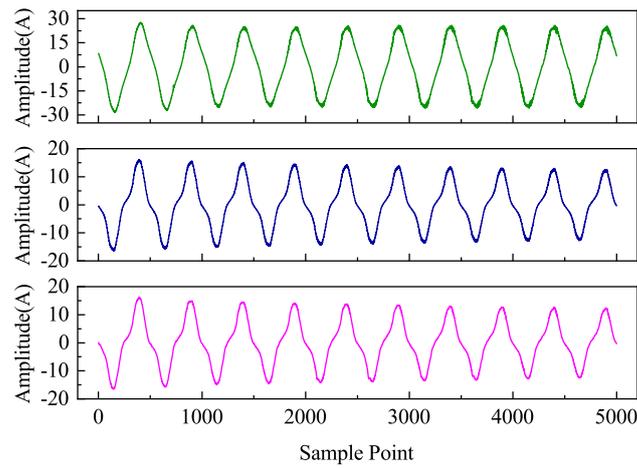


(b) Energy-saving lamps

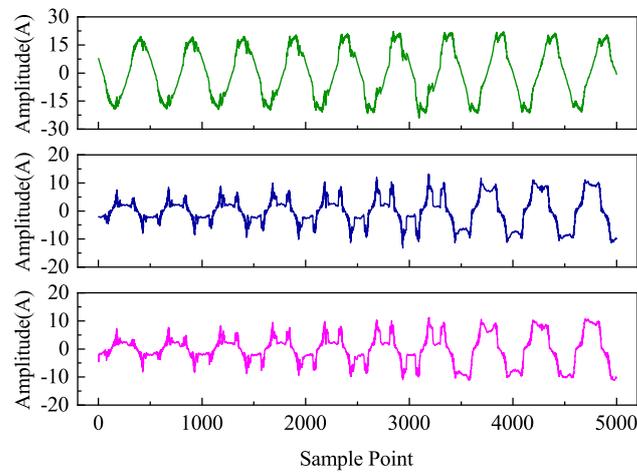


(c) Laptops

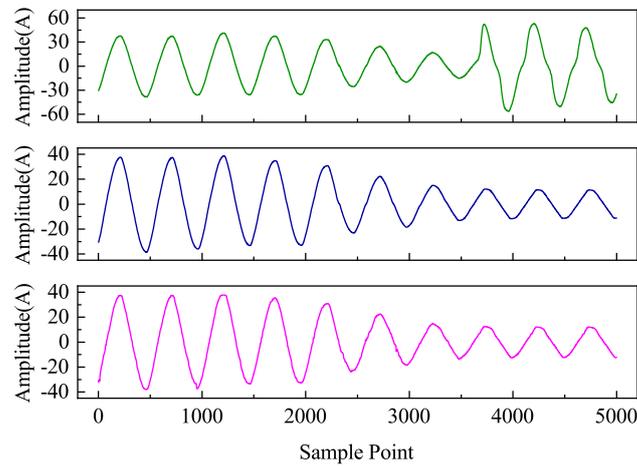
Figure 6. Cont.



(d) Vacuum cleaners



(e) Microwave ovens



(f) Washing machines

**Figure 6.** Monochromatic VI trajectory, VI trajectory with color coding, and color-coded VI trajectory with background colors.

**Table 3.** Evaluation metrics scores of the LATM-DAE algorithm.

Load Type	RMSE	MAE	Phase Error	Correlation Coefficient (%)
Air conditioner	0.109	0.081	0.068	99.8
Energy-saving lamps	0.033	0.020	0.196	98.1
Notebook	0.097	0.040	0.332	94.5
Vacuum cleaner	0.295	0.220	0.035	99.9
Microwave oven	0.668	0.456	0.129	99.2
Washing machines	0.866	0.551	0.046	99.9

### 5.2.2. Load Recognition

After converting the dataset into VI trajectory images with background color coding, it was input into the CNN network, and the Bayesian Optimization algorithm was used to optimize the hyperparameters of the convolutional layers, pooling layers, and Dropout layers. There were a total of five convolutional layers, three pooling layers, and two Dropout layers. The comparison of the hyperparameters before and after optimization is shown in Table 4. The data in the table indicate, for example, that the parameters of Conv1 layer were  $3 \times 3/48/1$ , which meant the convolutional kernel size was  $3 \times 3$ , the number of convolutional kernels was 48, and the stride was 1. The parameter of Dropout1 layer was 0.5, indicating a Dropout probability of 0.5.

**Table 4.** Comparison of the hyperparameter information between CNN and BOA-CNN.

Category	CNN	BOA-CNN
Conv1	$3 \times 3/48/1$	$4 \times 4/60/1$
Pool1	$3 \times 3/2$	$2 \times 2/1$
Conv2	$5 \times 5/128/2$	$4 \times 4/121/1$
Pool2	$3 \times 3/2$	$2 \times 2/1$
Conv3	$3 \times 3/192/1$	$4 \times 4/126/1$
Conv4	$3 \times 3/192/1$	$3 \times 3/55/1$
Conv5	$3 \times 3/192/1$	$3 \times 3/125/1$
Droout1	0.5	0.4
Droout1	0.5	0.1

The accuracy of the optimized model was compared with the recognition accuracy of other models in the literature, as shown in Table 5. The accuracy of the CNN network optimized by Bayesian Optimization was approximately 96.5%, while the accuracy of the regular CNN network was approximately 93.1%, indicating a 3.4% improvement. The recognition accuracy was also higher than that of other models in the literature.

**Table 5.** Comparison of the recognition accuracy of various algorithms.

Model	Accuracy (%)
BOA-CNN	96.5%
CNN	93.1%
Model in [39]	90.0%
Model in [19]	92.8%
Model in [40]	91.7%
Model in [41]	93.20%

In order to test the stability of the BOA-CNN algorithm, 12 experiments were conducted on the PLAID dataset. As shown in Figure 7, the recognition accuracy of the BOA-CNN algorithm was superior to that of the ordinary CNN model in all experiments, with good robustness. However, in terms of stability, the BOA-CNN algorithm performed poorly and fluctuated significantly. Specifically, in the 12 experiments, the difference between the maximum and minimum accuracy of the BOA-CNN algorithm was about 4.7%,

while the difference between those of the ordinary CNN model was about 3.5%. This indicates that the BOA-CNN algorithm is unstable across different experiments. Further analysis showed that this instability may be caused by the randomness and local optimal solution of Bayesian Optimization algorithms. However, its automatic search for the optimal hyperparameters of the convolution layer, pooling layer, and Dropout layer reduces the time and labor costs of manual parameter adjustment.

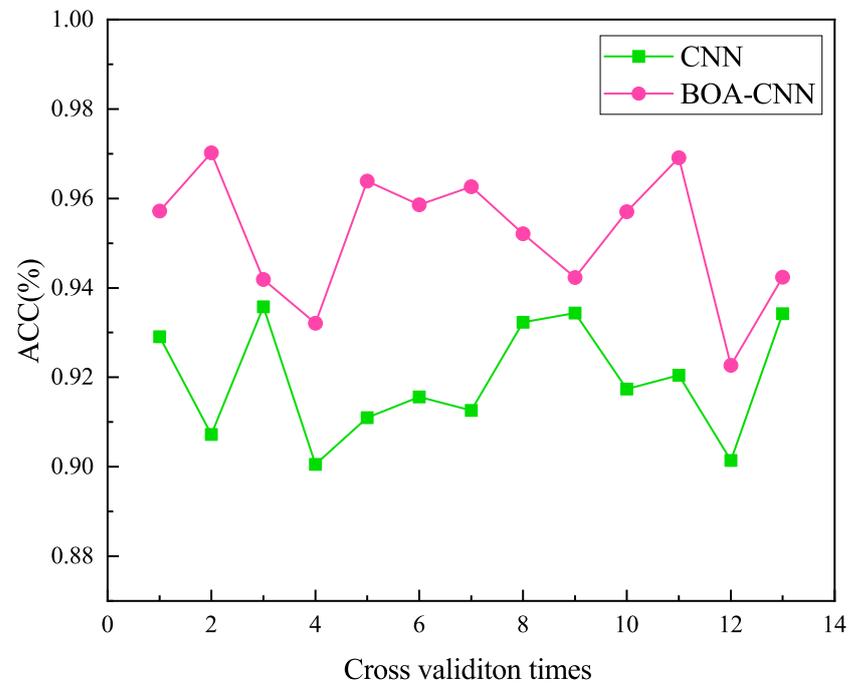


Figure 7. Comparison of the recognition accuracy between the BOA-CNN and CNN.

The model confusion matrix is shown in Figure 8, in which the number in each cell represents the number of corresponding devices, the abscissa represents the predicted value for an appliance, and the ordinate represents the actual value for that appliance. It can be seen that in the PLAID dataset, compared to the ordinary CNN model, the BOA-CNN model reduced the number of false negatives (FN) and false positives (FP) for each electrical appliance and also reduced the number of device types that were incorrectly classified for each electrical appliance. This indicates that the BOA-CNN model showed excellent results in the accuracy of electrical identification and has practical value.

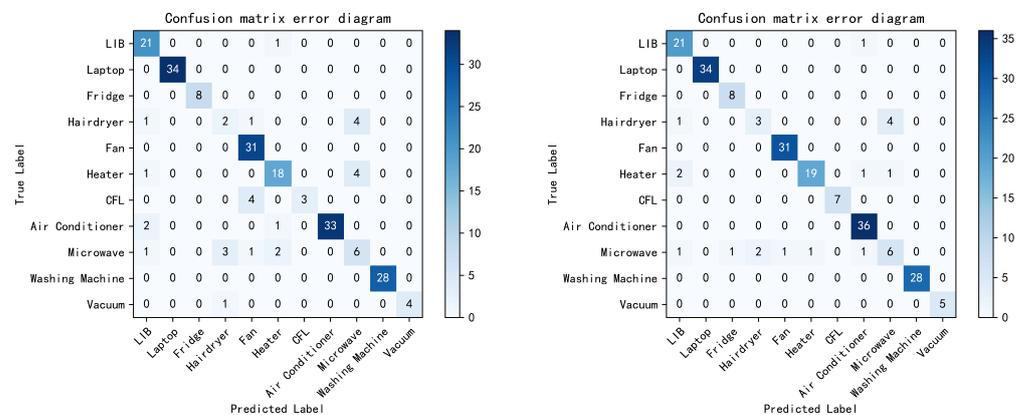


Figure 8. Comparison of the BOA-CNN and CNN confusion matrices.

In the case of the  $F$ -measure indicators and  $F_{macro}$  indicators, as shown in Figure 9, the  $F_{macro}$  indicator for the BOA-CNN was 87%, while the  $F_{macro}$  indicator for the CNN was 81%. In addition, the  $F$ -measure values for the hair dryers were lower than the macro average. This may be due to the small amount of measurement data for some devices in the dataset, which limits the learning ability of the model, resulting in lower  $F$ -measure values for some devices. However, in general, the convolutional neural network model based on Bayesian Optimization proposed in this paper showed improved performance compared to the original convolutional neural network model for 11 types of electrical appliances in the PLAID dataset, with a maximum improvement of 21%. Therefore, the model has good performance.

The incremental addition of numerous features during the CNN construction process led to a significant increase in computational workload. The experiment was conducted on a laptop equipped with 16 GB of RAM and 16 cores (12th Gen Intel(R) Core(TM) i5-1240P @1.70 GHz). On this platform, the time required for VI trajectory construction was 2016.47 milliseconds, while the recognition of input images took 868.01 milliseconds, resulting in a total of 2884.48 milliseconds, which is less than 3 s. In comparison, the total time spent in Reference [11] was 388.24 milliseconds, and in Reference [42], it was 923 milliseconds. Despite the relatively substantial computational workload, the CNN exhibits the capability to perform load recognition within seconds in the context of NILM tasks, aligning with the intended goals and requirements of recognition.

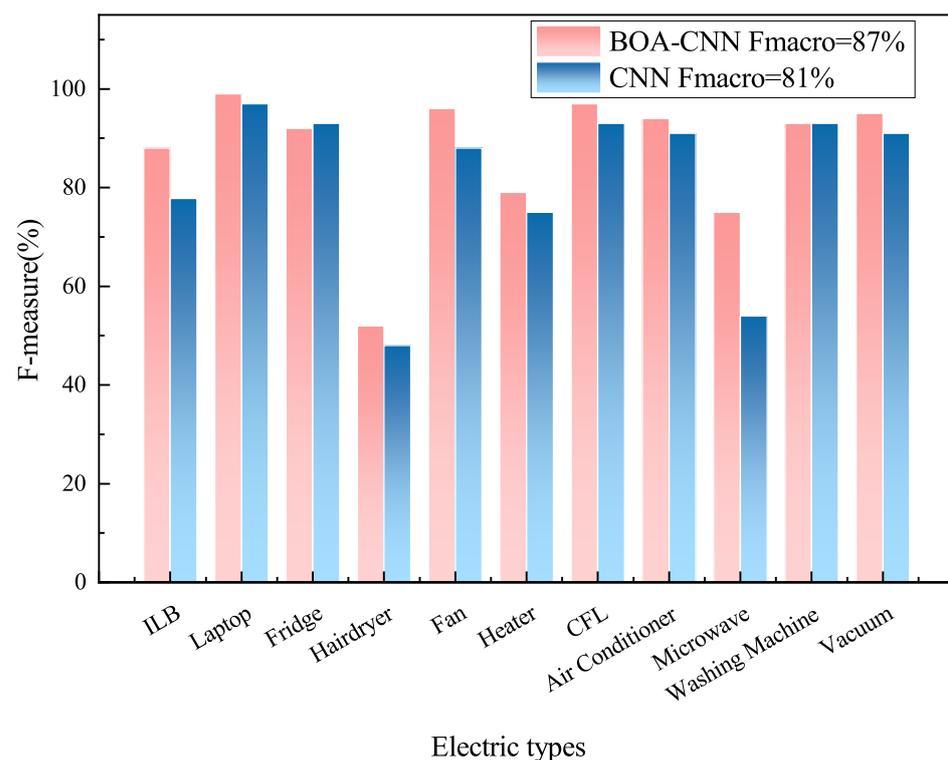


Figure 9. Comparison between the BOA-CNN and CNN  $F$ -measure indicators and  $F_{macro}$  indicators.

## 6. Conclusions

This paper introduced a load decomposition method based on the LSTM-DAE and presented a load recognition approach that employed color-coded VI trajectories with background color filling. The experimental results revealed that the proposed load decomposition method achieved an accuracy exceeding 98%, accompanied by a notable current reproduction effect, thereby providing precise data support for high-frequency load recognition. The amalgamation of multicycle signals, the color coding of VI trajectories, and the background color filling enhanced the distinguishability among different load VI trajectories. Moreover, the utilization of a BOA-optimized AlexNet network for

VI trajectory recognition illustrated that the BOA-CNN model effectively mitigated the issue of confusing devices compared to existing algorithms, showcasing elevated accuracy and robustness. Relative to conventional CNN models, this method excelled in various aspects of electrical appliance recognition. Although the algorithm presented in this paper imposed a higher computational load than other lightweight algorithms, the processing time aligned seamlessly with the requirements of load recognition. Nevertheless, despite the high accuracy of the proposed decomposition method, it required preliminary data training, unlike the differencing method, which does not require data pretraining. Consequently, the proposed decomposition method exhibited certain limitations compared to the differencing approach. Additionally, errors might have arisen in the CNN's recognition of VI trajectories due to significant differences among similar loads resulting from variations in brand and power. Therefore, future research should further explore the scalability of the load recognition capabilities of the proposed algorithm.

**Author Contributions:** methodology, D.Z. and J.S.; software, D.Z.; validation, J.S., R.F. and D.Z.; investigation, R.F.; resources, J.S.; data curation, D.Z. and R.F.; writing—original draft preparation, D.Z. and J.S.; writing—review and editing, J.S. and R.F.; visualization, D.Z.; supervision, R.F.; project administration, J.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** Shandong Jianzhu University Doctoral Fund (Project No.: X21103Z).

**Data Availability Statement:** Publicly available datasets were analyzed in this study. The data can be found here: [https://figshare.com/articles/dataset/PLAID\\_2014/11605074](https://figshare.com/articles/dataset/PLAID_2014/11605074).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Qi, B.; Liu, L.; Han, L.; Wang, L.; Ruan, W. Home appliance load identification algorithm based on system model. *Electr. Meas. Instrum.* **2018**, *55*, 23–30.
2. Deng, X.; Zhang, G.; Wei, Q.; Peng, W.; Li, C. A survey on the non-intrusive load monitoring. *Acta Autom. Sin.* **2022**, *43*, 644–663.
3. Li, P. Non-Intrusive Method for Power Load Disaggregation and Monitoring. Master's Thesis, Tianjin University, Tianjin, China, 2009.
4. Basu, K.; Debusschere, V.; Douzal-Chouakria, A.; Bacha, S. Time series distance-based methods for non-intrusive load monitoring in residential buildings. *Energy Build.* **2015**, *96*, 109–117. [[CrossRef](#)]
5. Ruano, A.; Hernandez, A.; Ureña, J.; Ruano, M.; Garcia, J. NILM Techniques for Intelligent Home Energy Management and Ambient Assisted Living: A Review. *Energies* **2019**, *12*, 2203. [[CrossRef](#)]
6. Piccialli, V.; Sudoso, A. A nonintrusive recognition method of household load behavior based on DTW algorithm. *Electr. Meas. Instrum.* **2019**, *56*, 17–22.
7. Zhu, H.; Cao, N.; Lu, H. Non-intrusive load identification method based on feature weighted KNN. *Electron. Meas. Technol.* **2022**, *56*, 70–75.
8. Dong, Z.; Chen, Y.; Xue, T.; Shao, R. Non-intrusive load monitoring algorithm based on attention mechanism combined with global and sliding window. *Electr. Meas. Instrum.* **2023**, *60*, 74–80.
9. Yu, D.; Liu, M. A non-invasive load decomposition method base on deep circular convolutional model. *Electr. Meas. Instrum.* **2020**, *57*, 47–53.
10. Athanasiadis, C.L.; Papadopoulos, T.A.; Doukas, D.I. Real-time non-intrusive load monitoring: A light-weight and scalable approach. *Energy Build.* **2021**, *253*, 111523. [[CrossRef](#)]
11. Papageorgiou, P.; Mylona, D.; Stergiou, K.; Bouhouras, A.S. A Time-Driven Deep Learning NILM Framework Based on Novel Current Harmonic Distortion Images. *Sustainability* **2023**, *15*, 12957. [[CrossRef](#)]
12. Lam, H.Y.; Fung, G.S.K.; Lee, W.K. A novel method to construct taxonomy electrical appliances based on load signatures. *IEEE Trans. Consum. Electron.* **2007**, *53*, 653–660. [[CrossRef](#)]
13. De Baets, L.; Develder, C.; Dhaene, T.; Deschrijver, D. Automated classification of appliances using elliptical fourier descriptors. In Proceedings of the 2017 IEEE International Conference on Smart Grid Communications, Dresden, Germany, 23 October 2017.
14. Hassan, T.; Javed, F.; Arshad, N. An Empirical Investigation of V-I Trajectory Based Load Signatures for Non-Intrusive Load Monitoring. *IEEE Trans. Smart Grid* **2014**, *5*, 870–878. [[CrossRef](#)]
15. Liu, Y.; Wang, X.; You, W. Non-intrusive load monitoring by voltage–current trajectory enabled transfer learning. *IEEE Trans. Smart Grid* **2018**, *10*, 5609–5619. [[CrossRef](#)]
16. Faustine, A.; Pereira, L.; Klemenjak, C. Adaptive weighted recurrence graphs for appliance recognition in non-intrusive load monitoring. *IEEE Trans. Smart Grid* **2020**, *12*, 398–406. [[CrossRef](#)]

17. Jia, D.; Li, Y.; Du, Z.; Xu, J.; Yin, B. Non-intrusive load identification using reconstructed voltage–current images. *IEEE Access* **2021**, *9*, 77349–77358. [[CrossRef](#)]
18. Han, Y.; Xu, Y.; Huo, Y.; Zhao, Q. Non-intrusive load monitoring by voltage–current trajectory enabled asymmetric deep supervised hashing. *Iet Gener. Transm. Distrib.* **2021**, *15*, 3066–3080. [[CrossRef](#)]
19. Han, Y.; Li, K.; Feng, H.; Zhao, Q. Non-intrusive load monitoring based on semi-supervised smooth teacher graph learning with voltage–current trajectory. *Neural Comput. Appl.* **2022**, *34*, 19147–19160. [[CrossRef](#)]
20. Kim, H.; Marwah, M.; Arlitt, M.; Lyon, G.; Han, J. Unsupervised Disaggregation of Low Frequency Power Measurements. In Proceedings of the 2011 SIAM International Conference on Data Mining, Mesa, AZ, USA, 28–30 April 2011.
21. Kolter, Z.; Jaakkola, T.; Kolter, J.Z. Approximate Inference in Additive Factorial HMMs with Application to Energy Disaggregation. *J. Mach. Learn. Res.* **2012**, *22*, 1472–1482.
22. Li, D.; Li, J.; Zeng, X.; Stankovic, V.; Stankovic, L.; Xiao, C.; Shi, Q. Transfer learning for multi-objective non-intrusive load monitoring in smart building. *Appl. Energy* **2023**, *329*, 1472–1482. [[CrossRef](#)]
23. Kelly, J.; Knottenbelt, W. Neural NILM: Deep Neural Networks Applied to Energy Disaggregation. In Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built, BuildSys 2015, Seoul, Republic of Korea, 4 November 2015.
24. Roberto, B.; Felicetti, A.; Principi, E. Denoising Autoencoders for Non-Intrusive Load Monitoring: Improvements and Comparative Evaluation. *Energy Build.* **2018**, *158*, 1461–1474.
25. Odysseas, K.; Christoforos, N.; Dimitris, V. Sliding Window Approach for Online Energy Disaggregation Using Artificial Neural Networks. In Proceedings of the 10th Hellenic Conference on Artificial Intelligence, Patras, Greece, 9 July 2018.
26. Zhang, C.; Zhong, M.; Wang, Z.; Goddard, N.; Sutton, C. Sequence-to-Point Learning with Neural Networks for non-intrusive Load Monitoring. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
27. Xu, X.; Zhao, S.; Cui, K. Non-intrusive load decomposition algorithm based on convolution block attention model. *Power Syst. Technol.* **2021**, *45*, 3700–3705.
28. Piccialli, V.; Sudoso, A. Improving Non-Intrusive Load Disaggregation through an Attention-Based Deep Neural Network. *Energies* **2021**, *14*, 847. [[CrossRef](#)]
29. Hart, G. Nonintrusive appliance load monitoring. *Proc. IEEE* **1992**, *80*, 1870–1891. [[CrossRef](#)]
30. Vincent, P.; Larochelle, H.; Bengio, Y.; Manzagol, P.A. Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 5 July 2008.
31. Liu, H.; Liu, Y.; Deng, S.C.; Shi, S.B.; Min, R.L.; Zhou, D.G. A power load identification method based on LSTM model. *Electr. Meas. Instrum.* **2019**, *56*, 62–69.
32. Han, T.; Chen, X.; Liu, Q. Short-Term Load Forecasting for Distribution Network in the Presence of TOU Price Based on Long-Short-Term Memory Network. *J. Northeast Electr. Power Univ.* **2020**, *40*, 19–28.
33. Song, X.; Jin, L.; Zhao, Y.; Yue, S.; Tong, L. Plant Image Recognition with Complex Background Based on Effective Region Screening. *Laser Optoelectron. Prog.* **2020**, *57*, 181–191.
34. Liu, F.; Li, M.; Hu, J.; Xiao, Y.; Qi, Z. Expression Recognition Based on Low Pixel Face Images. *Laser Optoelectron. Prog.* **2020**, *57*, 97–104.
35. Zhu, H.; Liu, X.; Liu, Y. Bayesian-based novel deep learning hyperparameter optimization. *Data Commun.* **2019**, *2*, 35–38.
36. Jones, D.; Schonlau, M.; Welch, W. Efficient global optimization of expensive black-box functions. *J. Glob. Optim.* **1998**, *13*, 455–492. [[CrossRef](#)]
37. Yu, X.; Xu, L.; Li, J.; Ji, X. MagConv: Mask-Guided Convolution for Image Inpainting. *IEEE Trans. Image Process.* **2023**, *32*, 4716–4727. [[CrossRef](#)]
38. Cui, H. Non-Intrusive Load Monitoring and Decomposition Technology Based on Deep Learning and Application of Typical Scenarios. Master’s Thesis, Northeast Electric Power University, Dalian, China, 2023.
39. Wu, X.; Jiao, D.; Liang, K.; Han, X. A fast online load identification algorithm based on VI characteristics of high-frequency data under user operational constraints. *Energy* **2019**, *188*, 63–70. [[CrossRef](#)]
40. Zhao, A.; Zhao, X.; Jing, J.; Xi, J.; Cui, P. Non-intrusive Electric Load Identification Algorithm for Optimizing CNN Hyperparameters. *Laser Optoelectron. Prog.* **2023**, *60*, 63–70.
41. Zhi, D.; Shi, J.; Fu, R. Non-intrusive Load Identification Based on Steady-state V-I Trajectory. In Proceedings of the 4th International Conference on Neural Computing for Advanced Applications, Hefei, China, 7 July 2023.
42. Wu, B.; Gu, W.; He, X. Low-power consumption high-precision non-invasive electrical appliance identification algorithm. *J. Xidian Univ.* **2023**, *50*, 149–157.

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