Non-intrusive load monitoring using CNN: A time-series data approach

Quang Son Ngo[1], Huy Hoang Nguyen[1], Tien Dung Do[1], Duc Chinh Hoang[1, *]

[1]School of Electrical and Electronic Engineering,
Hanoi University of Science and Technology
[*]Corresponding author E-mail:chinh.hoangduc@hust.edu.vn
DOI: https://doi.org/10.64032/mca.v29i4.323

Abstract

To ensure safe and comprehensive power monitoring within equipment limits, Non-Intrusive Load Monitoring (NILM) is a leading solution due to its convenience and cost-effectiveness. Various NILM approaches exist, including statistical methods, non-electrical variable integration, and machine learning models. However, these methods often fail to fully capture load signal characteristics, particularly current-voltage relationships. The Convolutional Neural Network (CNN) model is the most effective for monitoring and predicting load combinations using V-A (voltage-ampere) trajectory graphs. However, transient events can distort these trajectories, leading to incorrect predictions. This study enhances CNN performance by integrating time-series features to improve accuracy and handle switching events. Input data is structured as sequential images representing circuit states over time. A baseline model is designed, with multiple variations tested to analyze the correlation between parameters and accuracy. The study successfully applies a CNN model with time-series characteristics for load identification, even during switching events. Experimental results determine the most optimal model for practical applications.

***Keywords**: NILM, CNN, time-series characteristics*

1. Introduction

Monitoring and evaluating energy consumption behavior is a critical task in optimizing energy-efficient system operation. Load monitoring is a method for assessing the status of power consumption within an electrical system. The monitored quantities typically include current, voltage, and power of the loads in real time, allowing for the identification of load characteristics within the system [1].

Load monitoring is traditionally based on a multi-sensor model. In its initial stage, a sensor network model was employed as a primary approach to load monitoring, where each electrical device was monitored by an individual sensor node. Each sensor exclusively tracked a single load without signal overlapping from other devices, ensuring high reliability and clarity of the monitored data. However, this approach has significant drawbacks, including high costs and complexity in managing large-scale sensor networks. These limitations make intrusive load monitoring a less viable solution.

To address these challenges, a new approach is needed to minimize the number of measurement devices within the system. The concept of non-intrusive load monitoring (NILM) emerges as a method for measuring and monitoring load characteristics without requiring dedicated measurement devices for each individual load [2]. Instead, information about power consumption can be acquired using a single sensor. Figure 1 illustrates a measurement device positioned at the system's main power line to monitor the overall system's electrical parameters. All information regarding individual loads, despite their overlapping signals, is collected at a single monitoring point.

There are four fundamental load characteristics: capacitive, inductive, resistive, and electronic. Each electrical device combines these characteristics in a specific manner, forming its unique operational profile. However, in the context of non-intrusive load monitoring (NILM), the primary challenge is determining the operational state of individual devices using only the aggregated electrical data of the entire system.

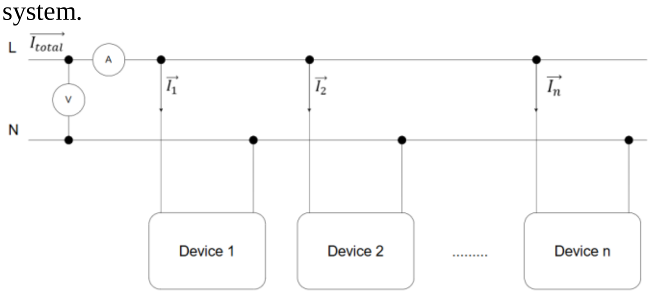

Figure 1: Basic system structure of non-intrusive load monitoring.

Y. F. Wong et al. proposed a hybrid approach that integrates electrical measurements with non-electrical features such as acoustic and optical signals emitted by the loads [3]. However, this method has proven to be less reliable due to the susceptibility of sound and light measurements to environmental variations. Currently, purely electrical data-based NILM is widely adopted for its robustness against external disturbances. The three main methodological approaches in this field include Machine Learning (ML), Pattern Matching (PM), and Single-Channel Source Separation (SS).

T. Hassan et al. successfully employed Support Vector Machines (SVMs) for NILM classification through feature extraction techniques [4]. Additionally, various simple machine learning models have demonstrated feasibility and promise in NILM applications. Notable approaches include Artificial Neural Networks (ANNs) [5], Decision Trees (DTs) [6], and Hidden Markov Models (HMMs) [7]. Numerous variations of these models have also been explored. However, these methods face accuracy limitations during signal preprocessing, as they rely heavily on statistical feature

Received: 03 April 2025; Revised: 15 June 2025; Accepted: 09 July 2025.

extraction. This raises concerns about the computational efficiency of machine-learning-based NILM solutions.

With advancements in deep learning and big data, feature extraction processes have been streamlined, reducing computational complexity. Consequently, researchers have shifted toward deep-learning-based NILM solutions. D. Murray et al. proposed using Convolutional Neural Networks (CNNs) and Gated Recurrent Units (GRUs) for load identification [8]. This approach eliminates intermediate feature extraction while maintaining high classification accuracy. Furthermore, the temporal aspect significantly impacts load state recognition. However, existing models have not yet incorporated time-sequence information to effectively handle switching events and system disturbances.

To address this limitation, researchers have introduced Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks to leverage sequential dependencies in NILM applications [9], [10], [11]. While these approaches exhibit strengths and weaknesses, they primarily focus on the characteristics of individual electrical measurements. However, a fundamental load characteristic often overlooked is the phase shift between current and voltage, which defines the system's Volt-Ampere (VA) graph.

In this study, CNNs are employed as the primary load identification models. Instead of raw electrical signals, the input data consists of time-series images of Volt-Ampere (V-I) graphs, which capture the temporal variations of the system. This approach allows for leveraging the geometric and frequency characteristics present in the images. The research also explores multiple 3D CNN model variants to compare their performance and select the most suitable model for load identification. This study presents the system framework, data acquisition principles, dataset construction methodology, and CNN training results.

NILM data, typically comprising voltage or current time series, contain distinctive activation patterns of appliances. CNNs efficiently capture such local patterns spikes, slopes, and sudden changes through convolutional layers, while significantly reducing the number of model parameters compared to fully connected networks, thereby mitigating overfitting and easing the training process. CNNs are particularly suitable for time series signal processing, offering faster and more stable training compared to recurrent neural networks (RNNs) and enabling practical deployment on resource-constrained devices. Moreover, CNNs can be integrated with LSTM, Transformer, or Attention mechanisms to enhance performance by combining local feature extraction with long-term dependency modeling. When compared to other NILM approaches, CNNs demonstrate a balanced trade-off between feature extraction capabilities, generalization, training efficiency, and deployment feasibility, making them highly effective for real-world applications

For equipment classification, there are two main techniques in load identification, one is to identify the on/off switches of the equipment on the system and the other is to identify the loads in the steady state. Professor Hart, who proposed the NILM concept, used the effective and reactive power as the identification features of the equipment. The steady state features include current waveform, current harmonics and V-I trajectory. The transient features include instantaneous power, instantaneous current waveform. In general, the transient state provides more distinguishable features than the steady state. However, the disadvantage of load identification in the transient state is that the sensor needs to be continuously monitored, so that no equipment is missed. In this study, we have extracted the features from the steady state [12].

2. Methodology

2.1 System description

The proposed Non-Intrusive Load Monitoring (NILM) system is designed to identify and monitor the operating states of electrical appliances in real time. As illustrated in Figure 2, the system comprises two main layers: the measurement hardware and the prediction software. The hardware layer is responsible for high-frequency acquisition and preprocessing of electrical signals, while the software layer handles data processing and load classification.

The hardware layer consists of several key components. The STM32 microcontroller collects current and voltage measurements using the ADS8361 16-bit simultaneous dual channel ADC at a 5 kHz sampling rate, enabling high-resolution signal capture. Current and voltage sensors provide raw electrical parameters from the main power line, which are processed through a signal preprocessing module to convert them into a differential format suitable for ADC processing. The Raspberry Pi 5 serves as the computational unit, running a pre-trained deep learning model for real-time load identification. An LCD display is used to visualize key parameters, including voltage, current, power, Total Harmonic Distortion (THD), and Fast Fourier Transform (FFT) analysis.

The software layer is responsible for data processing, feature extraction, and load classification. It consists of three main components. First, embedded software on the STM32 microcontroller, developed in C, acquires voltage-current signals, performs basic preprocessing, and transmits data to Raspberry Pi 5 via Ethernet. Second, a PC-based dataset collection software, developed in C#, is used to collect labeled load datasets for training the deep learning model. Finally, deep learning inference software on the Raspberry Pi 5, written in Python, processes measurement data using a Convolutional Neural Network (CNN) model. This model classifies active loads and transmits the results to the LCD screen for real-time monitoring.

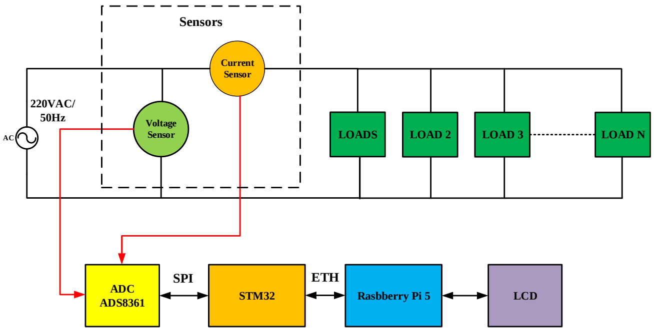

Figure 2: Functional diagram of the system

During operation, the system achieves real-time load monitoring and prediction with an accuracy of ≥90%, enabling users to analyze and optimize home energy usage. The STM32 microcontroller continuously acquires current-voltage signals and transmits them to Raspberry Pi 5, where the deep learning model identifies individual loads and predicts their operational states. The results are then displayed on the LCD screen, allowing users to monitor the electrical system efficiently. By integrating high-frequency data acquisition, advanced signal processing, and deep learning techniques, the system provides an effective and intelligent solution for real-time electrical load monitoring in smart buildings.

The process of collecting, processing data and classifying loads is carried out in the following steps:

- Step 1: The signal processing board will collect data on current and voltage in real time.
- Step 2: The data will be packaged into a message and transmitted to the Raspberry Pi5 module according to Ethernet standard.
- Step 3: The data will be processed and the V-I curve will be drawn.
- Step 4: The curve data will be passed through the trained 3D CNN model.
- Step 5: Classify the types of loads currently operating on the system.

2.2 Data Acquisition

2.2.1 *Data Collection and Signal Processing*

The system operates on a 220V AC/50Hz power supply, necessitating a signal preprocessing stage before analog-to-digital conversion (ADC) to ensure accurate voltage and current measurements. For current sensing, a Hall-effect sensor (TMCS1101A2BQDRQ1) with a measurement range of ±50A is employed. The sensor provides an output voltage within the range of 0V to 5V, corresponding to a current range of -50A to +50A. The Hall sensor serves two primary functions: isolating the AC measurement stage from the ADC stage and converting current signals into voltage signals suitable for ADC processing. Since the ADC input requires a differential signal within the 0V to 2.5V range, the Hall sensor output is further processed through a differential amplifier.

For voltage measurement, a transformer is used to step down the input voltage while simultaneously isolating the AC stage from the subsequent ADC measurement stage. After passing through the transformer, the voltage signal is processed using an INA128 instrumentation amplifier to apply an offset correction. This step is necessary because the measured voltage signal includes negative values, which must be shifted to an entirely positive voltage range before ADC conversion. The amplified signal is then fed into a differential amplifier to produce the final differential signal required by the ADC.

2.2.2 *Analog-to-Digital Conversion and Data Transmission*

The system employs the ADS8361 ADC, a 16-bit converter capable of high-frequency data acquisition. This ADC simultaneously captures voltage and current signals, with differential inputs minimizing noise interference. The central processing unit, an STM32 microcontroller, manages ADC data acquisition and communicates with the ADS8361 via an SPI interface.

After data acquisition, the collected measurements undergo preprocessing and packetization before transmission. To optimize data communication, a batch of 100 consecutive samples is aggregated before transmission, significantly reducing bandwidth consumption by minimizing initialization bytes and cyclic redundancy check (CRC) overhead. Instead of transmitting multiple smaller packets, this method consolidates data into a single larger packet, thereby reducing redundancy and improving transmission efficiency. The processed data is then transmitted to a Raspberry Pi 5 module for further analysis.

2.2.3 *Feature Extraction and Data Representation*

Figure 3: Lamp data chart: current, voltage, power, and FFT analysis

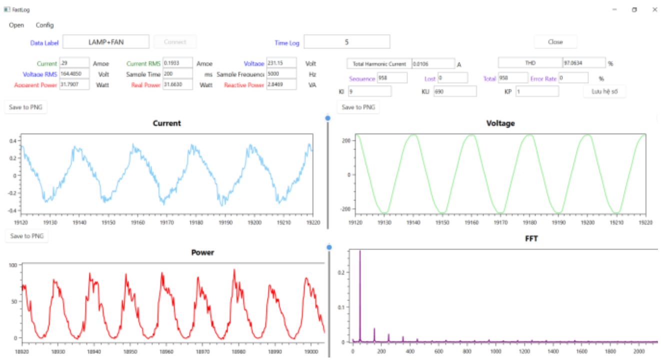

Figure 4: Lamp and fan data chart: current, voltage, power, and FFT analysis.

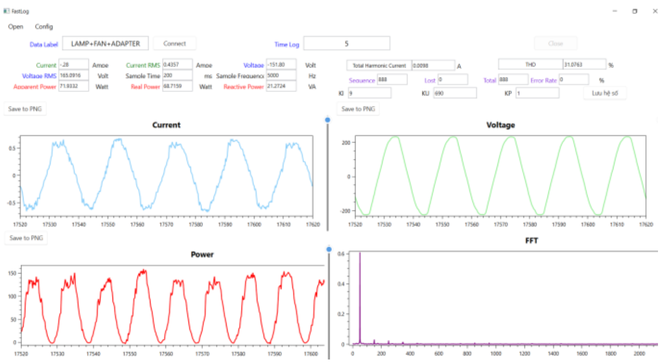

Figure 5: Data chart for 2 fans, 2 lamps, and 1 laptop adapter: current, voltage, power, and FFT analysis.

The Raspberry Pi 5 executes a pre-trained CNN-based model for load identification. The system utilizes extracted V-I trajectory features, transforming raw time-series data into a sequence of images representing the circuit's characteristic states over time. The dataset generated for model training consists of image sequences, each containing a predefined number of frames capturing the evolution of voltage and current waveforms.

CNN is a type of feedforward neural network that responds only to units in a local receptive field. It achieves local feature extraction through convolution operations. In addition, CNNs have fewer neurons, which prevents overfitting and allows deep features to be learned in a hierarchical manner. Furthermore, CNNs perform well in image processing tasks because they are invariant to observations such as rotation, translation, and scaling. In general, CNNs consist of convolutional layers, pooling layers, and fully connected layers. The convolutional layer consists of a number of convolutional filters (or kernels) and bias values. Convolutional filters are used to extract local features from an image, and different filters can extract different features. Filters slide over the image and perform convolutional operations, resulting in a variety of image features [13].

In addition to load identification, the system provides real-time visualization of essential electrical parameters such as current, voltage, power, Total Harmonic Distortion (THD), and FFT analysis for each detected load. This implementation ensures an efficient and accurate non-intrusive load monitoring (NILM) system, supporting advanced energy management and load disaggregation in complex electrical environments.

To illustrate this, Figure 3 depicts the electrical characteristics of a lighting system, while Figure 4 extends the analysis to both lighting and fan loads. Moreover, Figure 5 showcases a more intricate scenario involving multiple devices, demonstrating the system's capability in handling various electrical loads simultaneously.

2.3 Dataset

The dataset is derived from raw voltage and current data recorded in real time. These data are initially stored in an SQLite database before undergoing preprocessing to extract meaningful features for load identification. Given the dynamic nature of electrical loads, the dataset is designed to capture temporal variations in the voltage-current (V-I) characteristic curves.

To achieve this, a Python program is written to process the recorded data based on the sampling cycle to generate consecutive time-series images representing the V-I characteristic curves of different loads. These images effectively capture variations in load behavior over time, providing a rich feature set for classification. Since these images form sequential patterns, they can be interpreted as short video segments and are therefore well-suited for processing by a 3D CNN.

For each load type and combination, the raw database is processed to extract 200 data points per image, representing the V-I characteristic curve. Each input sample for CNN consists of a sequence of 10 images, corresponding to 2000 consecutive data points. These image sequences are stored in subfolders, each labeled with the corresponding load type and assigned a unique serial number for tracking. The images are saved in a 160×160 resolution to ensure uniformity and compatibility with the CNN model. Table 1 provides an overview of the loads and load combinations utilized in dataset creation for this study.

Table 1: Dataset of Loads Combinations

Main Load Group	Associated Loads in Combinations
Charger	Cooker, Fan + Cooker, Lamp + Cooker, Lamp + Fan, Monitor, Monitor + Cooker, Monitor + Fan, Monitor + Lamp
Cooker	None
Fan	Charger, Cooker, Lamp, Monitor
Lamp	Charger, Cooker, Fan + Cooker
Monitor	Cooker, Fan + Cooker, Fan + Lamp, Lamp, Lamp + Cooker
None	None

The power consumption of the devices is shown in Table 2. The above devices are connected to the power system by plugging into the common power outlet after the current and voltage sensors.

Table 2: Load power data

LOAD	Power
Fan	47W
Lamp	9W
Charger	65W
Cooker	450W
Monitor	35W

To facilitate model training and validation, the dataset is partitioned into two subsets: Train and Valid. A Python script randomly splits the subfolders into these directories, maintaining a 70%-30% ratio for each load type and combination. Following this, another script scans the dataset structure and generates two CSV files, train.csv and val.csv, listing all subfolder names along with their corresponding labels. During model training, these CSV files serve as an index, allowing the training pipeline to retrieve input image sequences and their labels efficiently.

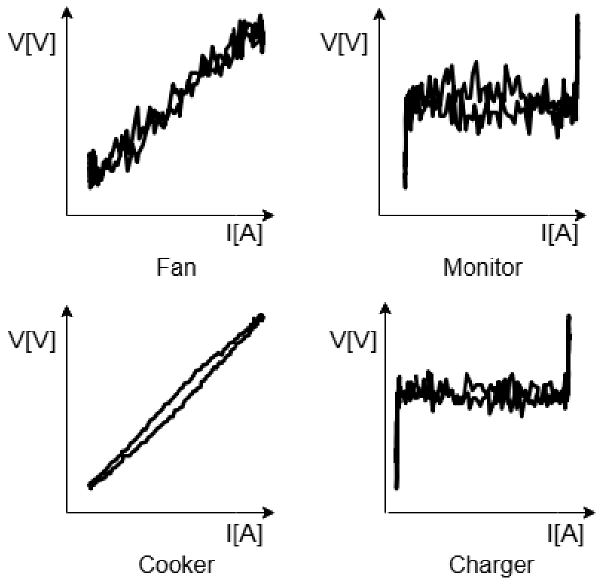

Figure 6: Sample V-I characteristic images of different loads

This structured dataset organization enables seamless data handling, simplifies debugging, and enhances reproducibility.

Figure 6 illustrates sample V-I characteristic images from different electrical loads, including a charger, a fan, and a lamp, highlighting the variations in their electrical signatures.

We use the raw data from the current and voltage sensors to generate a V-I feature image (Figure 6) from the collected data.

2.4 Convolution Neural Network

For a dataset consisting of sequences of images representing the voltage-current relationship at each cycle of the electrical circuit, the proposed CNN model is designed to recognize the characteristic patterns within these image sequences. As illustrated in Figure 7, the sample CNN architecture consists of three 3D convolutional layers with an increasing number of filters. In each convolutional layer, the feature maps obtained after convolution operations are processed through the ReLU activation function to eliminate negative values. To efficiently handle the large volume of image features, the outputs from the convolutional operations are batch-normalized before being passed to the next computational layer. A pooling layer is applied after each convolutional block to reduce the spatial dimensions of the feature maps, thereby enhancing computational efficiency.

After the input image sequence is processed through the three convolutional layers, the extracted features are flattened into a vector representation using a Flatten layer. This transformation prepares the data for subsequent fully connected layers, where the number of neurons gradually decreases across layers to optimize computational efficiency. The ReLU activation function is consistently applied across all hidden layers. Additionally, to enhance the model's generalization capability and prevent overfitting, a dropout mechanism is incorporated in each hidden layer, randomly deactivating a fraction of neurons during training. The final classification layer employs a Softmax activation function, with the number of neurons corresponding to the number of load combinations to be predicted. The model is trained using the cross-entropy loss function and optimized with the Adam algorithm to update network parameters efficiently.

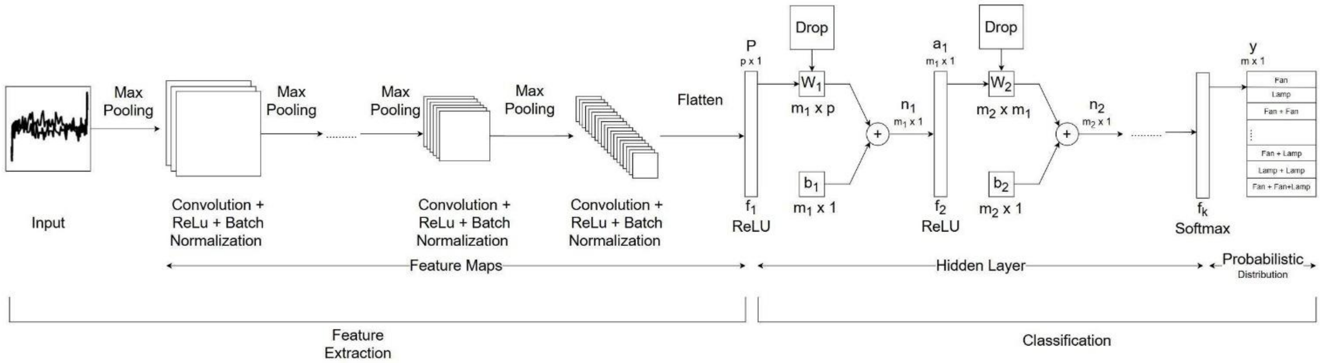

Figure 7: Sample proposed CNN model

3. Experimental Study and Results

3.1 Data Collection Process

The data collection process was meticulously designed to ensure high representativeness of electrical loads in real-world environments. A critical aspect of this process was the sampling frequency; electrical signals were recorded at 5 kHz to capture both transient and steady-state characteristics of the loads effectively. The choice of 5 kHz was made after careful consideration of the need to balance data resolution and storage constraints, ensuring that high-frequency transient events were captured without overwhelming computational resources during analysis.

Measurements were conducted using multiple approaches to build a diverse dataset that accurately reflects real-world conditions:

- Individual device operation: Each electrical appliance was measured independently to establish a baseline signature for its energy consumption, both in steady-state and during transient events such as power-on and power-off cycles.
- Simultaneous operation of similar devices: Multiple devices of the same type were operated concurrently to simulate potential interferences and overlapping load signatures.
- Combination of various devices: Different combinations of electrical appliances were measured to create realistic power consumption patterns, capturing the complexity of real-world electricity usage in homes and commercial buildings.
- Each measurement lasted for 10 seconds, ensuring sufficient data for model training and evaluation. This duration was chosen based on empirical analysis, as it provides an optimal trade-off between capturing short-term transient behaviors and reducing data redundancy over extended periods.

Figure 8: Load Recognition System

Figure 8 illustrates the Load Recognition System, which was used to process and analyze the collected data, ensuring accurate identification of electrical loads under various operational conditions.

3.2 Model Architecture Variants

The proposed models consist of multiple 3D convolutional architectures, each designed with different depths and filter configurations to optimize feature extraction for load recognition. Table 2 summarizes the key architectural components, including the number of convolutional layers, filter sizes, pooling strategies, dense layers, and learning rates for each model. These variants were developed to evaluate the trade-offs between computational complexity and classification performance.

Most models utilize four convolutional layers, with deeper architectures (e.g., ModelConv3D4) designed to enhance feature extraction. The primary filter size is (3,3,3), and MaxPooling3D (2,2,2) are applied for dimensionality reduction, though some models vary the pooling strategy to retain more information. Dropout layers help to prevent overfitting, improving generalization to unseen data. The Adam optimizer is used across all models, with a learning rate of 0.0002 for most cases, ensuring stable training behavior.

3.3 Model Training and Parameter Updates

The CNN-based NILM models were trained using a structured and systematic approach to ensure reliable classification of electrical loads. Stochastic Gradient Descent (SGD) with the Adam optimizer was employed due to its efficiency in handling large datasets and its capability to improve convergence speed by adapting learning rates dynamically. Categorical Cross-Entropy was chosen as the loss function because of its suitability for multi-class classification tasks, where the objective is to distinguish between different electrical appliances based on their unique load signatures.

Table 3: Overview of Model Architecture Variants

Model Name	Conv3D Layers	Filters per Layer	Filter Size	MaxPooling3D Strategy	Dense Layers	Learning Rate
ModelConv3D1	4	8 → 16 → 32 → 64	(3,3,3)	(2,2,2)/(1,2,2)	2	0.0002
ModelConv3D2	4	16 → 32 → 64 → 128	(3,3,3)	(2,2,2) / (1,2,2)	2	0.0002
ModelConv3D3	6	16 → 16 → 32 → 32 → 64 → 64	(3,3,3)	(2,2,2) / (1,2,2)	2	Default Adam
ModelConv3D4	8	16 → 16 → 32 → 32 → 64 → 64 → 128 → 128	(3,3,3)	(2,2,2) / (1,2,2)	2	Default Adam
ModelConv3D5	4	16 → 32 → 64 → 128	(3,3,3)/(2,2,2)	(2,2,2) / (1,2,2)	2	0.0002
ModelConv3D6	4	8 → 16 → 32 → 64	(3,3,3)/(2,2,2)	(2,2,2) / (1,2,2)	2	0.0002
ModelConv3D7	4	16 → 32 → 64 → 128	(3,3,3)	(2,2,2) / (1,2,2)	2	0.0002
ModelConv3D8	4	16 → 32 → 64 → 128	(3,3,3)	(2,2,2) / (1,2,2)	2	0.0002

As shown in Table 3, key training parameters and methodologies employed include:

- Initial learning rate: 0.0002, with dynamic adjustments based on validation performance. A learning rate scheduler was implemented to decrease the rate when model improvements plateaued, preventing overfitting and ensuring stable convergence.
- Batch size: Either 20 or 30, chosen based on hardware constraints and memory efficiency, ensuring that the models could generalize well without excessive computational burden.
- Epochs: Ranged from 3 to 30, depending on the complexity of the model variant. Shallower models required fewer epochs, while deeper models benefited from extended training to fully capture load-specific features.
- Optimization strategy: Weights were updated through backpropagation, ensuring that each iteration refined the model's ability to classify different electrical loads accurately. Dropout layers were incorporated across all models to mitigate overfitting, further improving generalization performance.

The initial learning rate of 0.0002 to ensure the model training process is stable and effective. With a fairly deep 3D-CNN network architecture, including many three-dimensional convolutional layers (Conv3D) combined with normalization layers (BatchNormalization), noise reduction (Dropout) and nonlinear activation, choosing a learning rate that is too high can lead to instability in the optimization process, causing the model to not converge or converge incorrectly.

On the other hand, a learning rate that is too low makes the training process slow, time-consuming and resource-consuming. Therefore, we choose the value of 0.0002 as a balance point - small enough to avoid strong fluctuations, but still large enough to ensure a reasonable learning rate.

Furthermore, this learning rate is used in combination with the Adam optimizer, which is an adaptive optimizer that can adjust the learning rate for each parameter. In many studies related to image and video sequence analysis using 3D-CNN networks, learning rates in the range of 0.0001 to 0.0005 are quite common.

Model Performance and Trade-offs

After training the models on the same hardware platform, we analyze performance based on accuracy, training time, and computational cost. The results show that all models achieve high accuracy, with the best model reaching 99.8%. However, architectural complexity directly impacts computational efficiency:

1. Highest accuracy model: Achieves 99.8% accuracy but has the highest number of parameters (3,644,378), leading to a longer training time (3167.05 seconds per epoch).
2. Efficient model: A lightweight model with 506,704 parameters maintains 99.5% accuracy while significantly reducing training time to 1718.05 seconds per epoch.

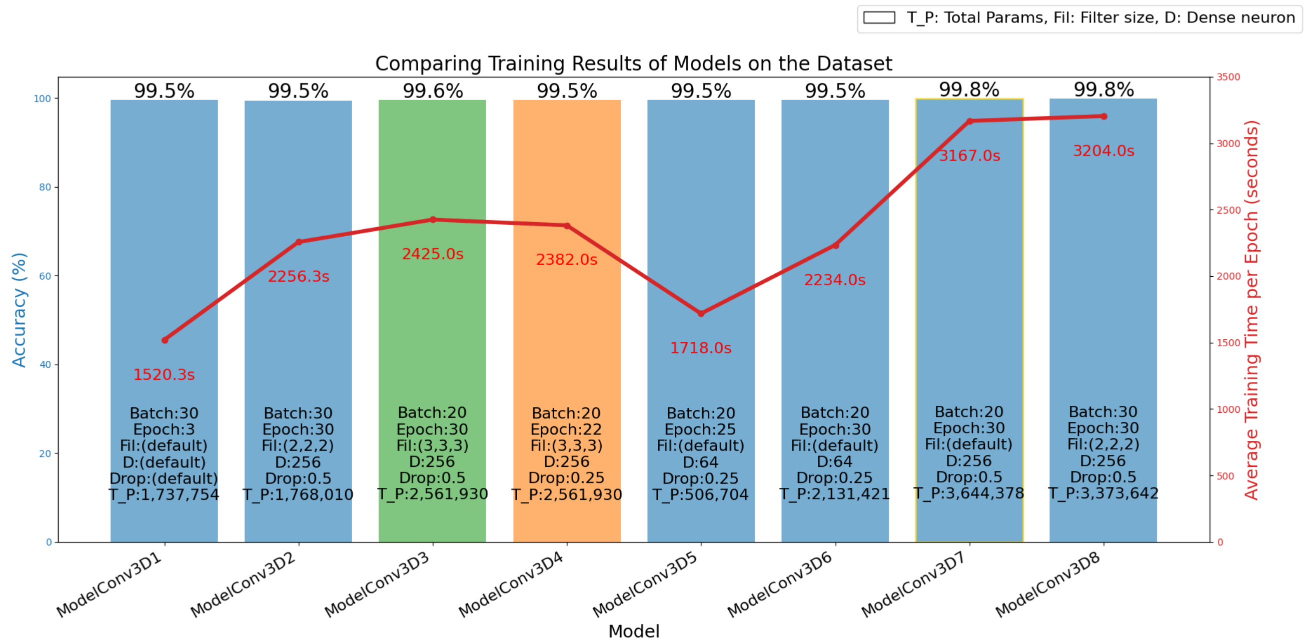

Figure 9: Models' parameters and accuracy comparing

These findings highlight the trade-off between model complexity and computational performance. While deeper networks can extract more features, they also require higher computational costs, which may not always be justified by a marginal improvement in accuracy. Figure 9 illustrates a comparison of Conv3D models in terms of accuracy, training time, and parameter count, providing a visual representation of the strengths and limitations of each architecture.

Looking at the chart in Figure 9, ModelConv3D7 provides the most accurate load predictions. The predicted load results using ModelConv3D7 are presented in Figure 10, demonstrating the model's ability to accurately identify the characteristics of electrical loads.

Figure 10: Predicted results of ModelConv3D7

Figure 10 is the fan load prediction result based on ModelConv3D7 using Python code with an accuracy of 0.9792. The images on the main screen are the V-I characteristics of the fan load.

We collected additional data, processed it, and combined it with the existing data to create new scenarios for accuracy testing. A total of eight trials were conducted, and the results are shown in the table 4 below.

Table 4: Accuracy of the models

Model	Accuracy (Mean ± SD) [%]	Training Time (Mean ± SD) [s]
ModelConv3D1	99.54 ± 0.11	1525.3 ± 10.95
ModelConv3D2	99.62 ± 0.09	2260.1 ± 15.22
ModelConv3D3	99.65 ± 0.12	2428.7 ± 12.55
ModelConv3D4	99.51 ± 0.10	2385.4 ± 14.88
ModelConv3D5	99.53 ± 0.08	1720.5 ± 9.77
ModelConv3D6	99.56 ± 0.11	2238.2 ± 11.33
ModelConv3D7	99.79 ± 0.05	3170.9 ± 8.44
ModelConv3D8	99.78 ± 0.07	3208.6 ± 13.11

We have conducted a concise comparative analysis highlighting the methodological and functional differences between our approach and selected representative NILM studies. While several works such as Liu et al. (2021) and Xinran Liu et al. (2024) utilize 3D CNNs and V–I trajectory features for appliance classification, they primarily focus on identifying single active devices during steady-state conditions [14], [15]. Our method advances this by addressing a multi-label classification task, enabling simultaneous recognition of overlapping appliance combinations.

Furthermore, we incorporate high-frequency V–I signals transformed into time-series images, allowing the network to capture spatiotemporal dynamics and transient behaviors.

Compared to Monteiro et al. (2021), who assess several ML techniques (e.g., MLP, CNN, LSTM) on low-frequency data for per-appliance classification, our use of high-frequency waveform data provides richer information content, enhancing discrimination performance [16]. In addition, while Medeiros et al. (2019) apply CNNs for event detection using low-frequency active/reactive power signals, our approach offers a finer resolution by learning device-level patterns from raw waveforms directly [17] .

Collectively, these distinctions particularly the focus on multi-label learning, transient robustness, and waveform-based feature representation position our method as a complementary and effective solution for NILM scenarios involving dense, overlapping appliance usage. We have integrated this comparison into the revised manuscript for clarity and completeness.

According to Liu et al. (2024), for NILM algorithms, a sampling frequency of less than 1 kHz is considered low-frequency sampling, typically applied in systems with few active loads and where high accuracy is not required. A sampling frequency of 1 kHz or higher is classified as high-frequency sampling, commonly used in systems with many active loads, requiring high accuracy and real-time monitoring. A sampling frequency of 5 kHz falls within the high-frequency range and is regarded as high-quality sampling, suitable for analyzing both harmonic features and rapid transient characteristics [18].

Furthermore, according to Hu et al. (2021), a sampling frequency between 4 kHz and 6 kHz is sufficient to distinguish the characteristic signatures of typical household appliances [19]. Additionally, the IEEE 519 standard (Recommended Practice and Requirements for Harmonic Control in Electric Power Systems) requires harmonic control up to at least the 50th order, which corresponds to a frequency of 2500 Hz. Following the Nyquist theorem, the minimum required sampling frequency is thus 5000 Hz (5 kHz) [20] .

In my NILM-based load identification application, the load characteristics do not require monitoring of very high-order harmonics. Moreover, a 5 kHz sampling frequency ensures that the data volume remains manageable, preventing system overload, while still maintaining measurement accuracy.

Besides, we also try higher frequencey at 10 kHz and 15 kHz. Then, we fed the collected data to train the model and ran the model to verify the results (as shown in table 5).

Table 5: Accuracy of system at different sampling frequencies

No	Sampling frequency	Accuracy
1	5 Khz	99.51
2	10 Khz	99.72
3	15 Khz	99.56

From the data table above, we see that higher sampling frequencies also give results with the same accuracy as the 5 Khz sampling frequency. The 5 Khz sampling frequency has ensured the necessary information to train the model. At the same time, with the 5 Khz sampling frequency, the amount of data will not be too large, ensuring that the system can run for a long time with a hardware system that does not need to be too strong.

4. Conclusion

This research establishes a solid foundation for NILM systems in practical settings. By balancing architectural complexity with computational efficiency, the proposed CNN-based approach effectively recognizes electrical loads, achieving high classification accuracy. The best-performing model attained 99.8% accuracy, demonstrating strong applicability in both residential and industrial environments. Additionally, the model efficiently captures transient and steady-state load characteristics, enabling robust recognition even under overlapping signatures and rapid switching conditions.

Despite its strong performance, the model has two key limitations. First, handling unknown loads remains a challenge, as newly introduced devices require retraining to be properly classified. Second, generalization across diverse environments is constrained by the dataset's representativeness. Future work will focus on adaptive learning techniques, such as incremental training and anomaly detection, to improve flexibility. Additionally, optimizing real-time deployment via edge computing and model quantization will enhance practical usability. By addressing these challenges, the framework will be better suited for energy auditing, fault detection, and demand response systems, bridging the gap between controlled experiments and real-world applications.

References

[1] C. Laughman *et al.*, "Power signature analysis," *IEEE Power Energy Mag.*, vol. 1, no. 2, pp. 56–63, Mar. 2003, doi: 10.1109/MPAE.2003.1192027.

[2] G. W. Hart, "Nonintrusive appliance load monitoring," *Proc. IEEE*, vol. 80, no. 12, pp. 1870–1891, Dec. 1992, doi: 10.1109/5.192069.

[3] Y. F. Wong, Y. Ahmet Şekercioğlu, T. Drummond, and V. S. Wong, "Recent approaches to non-intrusive load monitoring techniques in residential settings," in *2013 IEEE Computational Intelligence Applications in Smart Grid (CIASG)*, Apr. 2013, pp. 73–79. doi: 10.1109/CIASG.2013.6611501.

[4] T. Hassan, F. Javed, and N. Arshad, "An Empirical Investigation of V-I Trajectory Based Load Signatures for Non-Intrusive Load Monitoring," *IEEE Trans. Smart Grid*, vol. 5, no. 2, pp. 870–878, Mar. 2014, doi: 10.1109/TSG.2013.2271282.

[5] Y.-H. Lin and M.-S. Tsai, "An Advanced Home Energy Management System Facilitated by Nonintrusive Load Monitoring With Automated Multiobjective Power Scheduling," *IEEE Trans. Smart Grid*, vol. 6, no. 4, pp. 1839–1851, Jul. 2015, doi: 10.1109/TSG.2015.2388492.

[6] P. Bilski and W. Winiecki, "Generalized algorithm for the non-intrusive identification of electrical appliances in the household," in *2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, Sep. 2017, pp. 730–735. doi: 10.1109/IDAACS.2017.8095186.

[7] L. Mauch and B. Yang, "A novel DNN-HMM-based approach for extracting single loads from aggregate power signals," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2016, pp. 2384–2388. doi: 10.1109/ICASSP.2016.7472104.

[8] D. Murray, L. Stankovic, V. Stankovic, S. Lulic, and S. Sladojevic, "Transferability of Neural Network Approaches for Low-rate Energy Disaggregation," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 8330–8334. doi: 10.1109/ICASSP.2019.8682486.

[9] W. He and Y. Chai, “An Empirical Study on Energy Disaggregation via Deep Learning,” presented at the 2016 2nd International Conference on Artificial Intelligence and Industrial Engineering (AIIE 2016), Atlantis Press, Nov. 2016, pp. 338–342. doi: 10.2991/aiie-16.2016.77.

[10] İ. H. Çavdar and V. Faryad, “New Design of a Supervised Energy Disaggregation Model Based on the Deep Neural Network for a Smart Grid,” *Energies*, vol. 12, no. 7, p. 1217, Mar. 2019, doi: 10.3390/en12071217.

[11] L. Mauch and B. Yang, “A new approach for supervised power disaggregation by using a deep recurrent LSTM network,” in *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Dec. 2015, pp. 63–67. doi: 10.1109/GlobalSIP.2015.7418157.

[12] X. Liu, D. Liao, X. Zhang, and H. Gao, “Appliance Classification by Using Dynamic Voltage-Current Trajectory and 3D-Convolutional Neural Network,” in *2024 Boao New Power System International Forum - Power System and New Energy Technology Innovation Forum (NPSIF)*, Dec. 2024, pp. 468–474. doi: 10.1109/NPSIF64134.2024.10883241.

[13] R. Ma, W. Kang, J. Sha, J. Liu, and D. Zhu, “NILM Load Identification Based on CNN and V-I Trajectory,” in *2024 IEEE 2nd International Conference on Power Science and Technology (ICPST)*, May 2024, pp. 2079–2084. doi: 10.1109/ICPST61417.2024.10601881.

[14] J. Liu, N. Liu, H. Song, X. Liu, X. Sun, and D. Zhang, “Non-Intrusive Load Identification Model Based on 3D Spatial Feature and Convolutional Neural Network,” *Energy Power Eng.*, vol. 13, no. 04, pp. 30–40, 2021, doi: 10.4236/epe.2021.134B004.

[15] X. Liu, D. Liao, X. Zhang, and H. Gao, “Appliance Classification by Using Dynamic Voltage-Current Trajectory and 3D-Convolutional Neural Network,” in *2024 Boao New Power System International Forum - Power System and New Energy Technology Innovation Forum (NPSIF)*, Qionghai, China: IEEE, Dec. 2024, pp. 468–474. doi: 10.1109/NPSIF64134.2024.10883241.

[16] R. V. A. Monteiro, J. C. R. De Santana, R. F. S. Teixeira, A. S. Bretas, R. Aguiar, and C. E. P. Poma, “Non-intrusive load monitoring using artificial intelligence classifiers: Performance analysis of machine learning techniques,” *Electr. Power Syst. Res.*, vol. 198, p. 107347, Sep. 2021, doi: 10.1016/j.epsr.2021.107347.

[17] A. P. Medeiros, L. N. Canha, D. P. Bertineti, and R. M. De Azevedo, “Event Classification in Non-Intrusive Load Monitoring Using Convolutional Neural Network,” in *2019 IEEE PES Innovative Smart Grid Technologies Conference - Latin America (ISGT Latin America)*, Gramado, Brazil: IEEE, Sep. 2019, pp. 1–6. doi: 10.1109/ISGT-LA.2019.8895291.

[18] Y. Liu, Y. Wang, and J. Ma, “Non-Intrusive Load Monitoring in Smart Grids: A Comprehensive Review,” 2024, *arXiv*. doi: 10.48550/ARXIV.2403.06474.

[19] M. Hu, S. Tao, H. Fan, X. Li, Y. Sun, and J. Sun, “Non-Intrusive Load Monitoring for Residential Appliances with Ultra-Sparse Sample and Real-Time Computation,” *Sensors*, vol. 21, no. 16, p. 5366, Aug. 2021, doi: 10.3390/s21165366.

[20] “IEEE SA - IEEE 519-2022.” Accessed: Jun. 09, 2025. [Online]. Available: https://standards.ieee.org/ieee/519/10677/