Computers, Materials & Continua
DOI: 10.32604/cmc.2022.028358
Article

Tech Science Press

Exploring CNN Model with Inrush Current Pattern for Non-Intrusive Load Monitoring

Sarayut Yaemprayoon and Jakkree Srinonchat*

Faculty of Engineering, Rajamangala University of Technology Thanyaburi, Pathum Thani, 12110, Thailand
*Corresponding Author: Jakkree Srinonchat. Email: jakkree.s@en.rmutt.ac.th

Received: 08 February 2022; Accepted: 05 May 2022

Abstract: Non-Intrusive Load Monitoring (NILM) has gradually become a research focus in recent years to measure the power consumption in households for energy conservation. Most of the existing algorithms on NILM models independently measure when the total current load of appliances occurs, and NILM usually undergoes the problem of signatures of the appliance. This paper presents a distingue NILM design to measure and classify the appliances by investigating the inrush current pattern when the alliances begin. The proposed method is implemented while the five appliances operate simultaneously. The high sampling rate of field-programmable gate array (FPGA) is used to sample the inrush current, and then the current is converted to be image patterns using the kurtogram technique. These images are arranged to be four groups of data set depending on the number of appliances operating simultaneously. Furthermore, the five proposed modifications convolutional neural networks (CNN), which is based on very deep convolutional networks (VGGNet), are designed by adjusting the size to decrease the training time and increase faster operation. The proposed CNNs are then implement as a classification model to compare with the previous models. The F1 score and Recall are used to measure the accuracy classification. The results showed that the proposed system could be achieved at 99.06 accuracy classification.

Keywords: Non-instructive load monitoring; kurtogram image; convolutional neural network; deep learning

1 Introduction

Smart Home is a technology for energy consumption control in electrical appliances, and this technology can increase the efficiency of better energy allocation. The examination process of appliance load is based on real-time [1–3]. However, system development that can identify types and energy management must consider the specification of a particular appliance, e.g., the motor as a critical component of an electrical appliance [4] or the heating element inside a heater [5]. It is challenging to study these variables. Therefore, the development of NILM is an exciting option [1,3,6,7]. The development of NILM requires an understanding of three key components, i.e., data acquisition, appliance classification, and energy analysis [8–10]. According to studies, NILM usually undergoes the problem of signatures of the appliance (caused by the usage of appliances in terrain) [11], particular problems of each appliance (signatures obtained by the operation of electrical appliances) [1,12,13], domains [14,15], and graph [16]. It also undergoes the operational problems of appliances, divided into four types [10]. 1) ON/OFF state of appliances [17,18]: It refers to binary operation status, e.g., lamp by their switches, with short operational time. It is barely feasible for differentiation. 2) Multi-state appliances [19]: The operational status of appliances changed with usage, e.g., the operation of kettles changes with adjusted values. It is challenging to understand signatures in any form of variables. Therefore, the analysis primarily relies on observation of their forms for a while to make sure of their clarity. 3) Variable power appliances [20]: The operational status of appliances changes continuously, e.g., the energy consumption of washing machines varies during washing/rinsing/steam. Users have to understand properties and observe the operation in the long run. 4) Constant consumption appliances [10,21]: Appliances operate all the time, e.g., refrigerators or air-conditioners operate periodically. Besides, there are studies to change signatures of appliances by trans-forming data of operation (wave, voltage, and current) into images [13,22], which describes VI-trajectory with CNN, tested by plug load appliance identification dataset (PLAID) and worldwide household and industry transient energy dataset (WHITED). Baets et al. [23], describes VI-trajectory with siamese neural networks (SNN), tested by PLAID, WHITED, and others [24,25]. After that, transform binary images into gray images was initiated to increase the signatures' features, such as [26], which presents dimensionally aligned signal projections (DASP) transformation up to unintended radiated emissions (URE) detection. Yang et al. [27], describes zero filling areas that added the features as single load identification and multi-load identification. Liu et al. [13], de-scribes color encoding by adding an HSV plane into the data of appliances. Transfers learning is applied with PLAID and WHITED datasets. Li et al. [28], describes the transformation of appliance signatures as a current time-frequency feature. It is to transform electrical signal features into spectrums. The test is up to PLAID and smart electricity dataset of ocean university of china dataset. The development of classification techniques is another principle of NILM development. According to related studies, it was found that deep learning techniques are prevalent [29,30], e.g., recurrent neural network (RNN) [31,32], long short-term memory (LSTM) [33,34], and CNN [12–13,25]. New models are also developed, e.g., long short-term memory-probabilistic neural network (LSTM-PNN) [21], AlexNet [35], dilated residual network (DRN) [36] , dilated residual attention network (DRAN) [37], or bidirectional dilated residual (Bi-TCN) [38]. All of these are studies with different goals. However, deep learning techniques usually undergo problems due to network sizes, resulting in huge disadvantages on training time. Therefore, transfer learning was brought for development to avoid negative transfer of training [29].

Despite satisfactory results from previous studies, there are still some drawbacks, e.g., detection methods of appliance signatures [39,40], binary signatures of appliance [22], and deep learning methods for developing appliance classification techniques. This research focuses on modifying the appliances signatures by transforming voltage appliances signatures into kurtogram, a method that relies on a spectrum as a mediator with the identity of the appliance as the domain. It also aims to modify CNN in terms of hierarchy and to adjust training parameters to enhance model efficiency.

2 Related Work

NILM techniques are recognized for many applications, such as home energy management systems and ambient assisted living. It recently can classify the appliances using the feature extracting technique which extracts energy data form, and then it labels the pattern of energy applied to each appliance class. The previous works presented a method for feature extraction that was used by checking the device on/off [41,42]. On the bayesian optimization and robustness of event detection methods in NILM. This method used a low sampling rate at 1 Hz, and its disadvantage is that it can only detect high-power appliances. It also takes a long time to inspect the energy generated. The high sampling rates were used to collect data with the FPGA to solve this problem. It can analyze complex data such as voltage-current curve characterization [12,22,23,43,44]. Harmonic analysis [18,43] frequency component analysis [14,22] from the steady-state of current and voltage signal, and a transient analysis [9] of the current signal occurring while the device is turned on. These features are then transferred to training models and classifiers in many techniques such as support vector machine [45], decision tree classifier [46], k-nearest neighbor [47], or artificial neural network [7].

The CNN is a class of artificial neural networks, most commonly applied to analyze visual imagery such as RNN network [31,32], LSTM Network [33,34], and CNN [12–13,25]. Also, new models are developed, for example, LSTM-PNN [21], DRN [36], DRAN [37], or Bi-TCN [38]. It is, moreover, applied to operate as deep learning techniques, for example, AlexNet [48], VGG16 [49], etc. Recently the CNN [13–15,21,28–32,35–38] is applied to be feature extraction and classification in order to the extraction function operates inside the convolution model. Previous research has shown a wide range of input formats and conversion formats. For example, Zhao et al. [16], recorded the electricity consumption data of each device for 8 min and resampled it to 599 × 1 data as input to CNN, but the problem with this work is that it takes a long time to process. This approach is proposed by reducing the recording time to 1 min [6]. However, it still takes a long time to record the data. Moreover, high-frequency sampling is approached to reduce the time process. It used only 150 mS of data feature by sampling current at the steady-state operation and then rearranging 80 × 1 data as input to the CNN [29]. However, the CNN is implemented by converting the input feature into 2D or image data [30,36], and it still requires the input data as a minute long. The 2D converting technique is explored to reduce input feature data collection, such as using the voltage current curves as inputs for CNN [12,23], steady-state current conversions using gramian angular field matrices algorithm to work with VGG16 Network etc [28]. Also, the kurtogram technique that uses to convert data from 1D to 2D [50,51] is explored, such as analyzing rolling bearings [52]. Therefore, the kurtogram technique proposes to convert the current signal, which collects from the high sampling rate, to be 2D data input for the deep learning technique in CNN. However, deep learning techniques usually undergo problems due to network sizes and the disadvantages of training time. This research proposes the load recognition technique for NILM. The kurtogram is operated to convert the transient signal of electrical appliances, and then the VGGNet is modified to identify the electrical appliances.

3 Methodology and Experiment Setup

The proposed method to develop NILM requires three major key components; 1) data collection, 2) data preprocessing, and 3) classification. Firstly, the data collection consists of load, which has five appliances; 1) Air condition 7, 033 Watt, 2) Air condition 3, 516 Watt, 3) Lamp 28 Watt, 4) Microwave 800 Watt, and 5) Pump 150 Watt. All the current loads are collected using the current sensor based on the high sampling rate of FPGA, which operates as data acquisition. The current loads are transferred to be digital form then passed through to data processing. Data processing is the second process that transforms the current digital format in 1D to 2D as an image by using the kurtogram technique. These image patterns then pass through to the CNN model, which is the final step of the proposed method. The CNN is modified based on the VGG model. Fig. 1., shows the process of the proposed method.

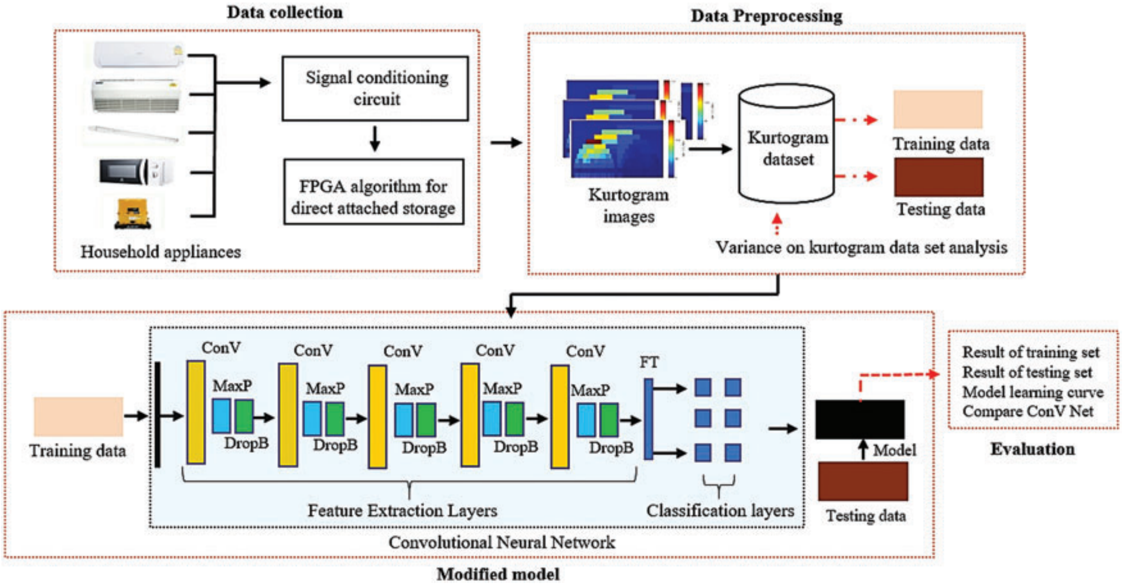


Figure 1: The process of the proposed method

3.1 Load and Inrush Current

According to studies, NILM usually undergoes the problem of signatures of the appliance (caused by the usage of appliances in terrain) [11], particular problems of each appliance (signatures obtained by the operation of electrical appliances) [1,12,13], domains [14,15], and graph [16]. It also undergoes the operational problems of appliances, divided into four categories [10,17].

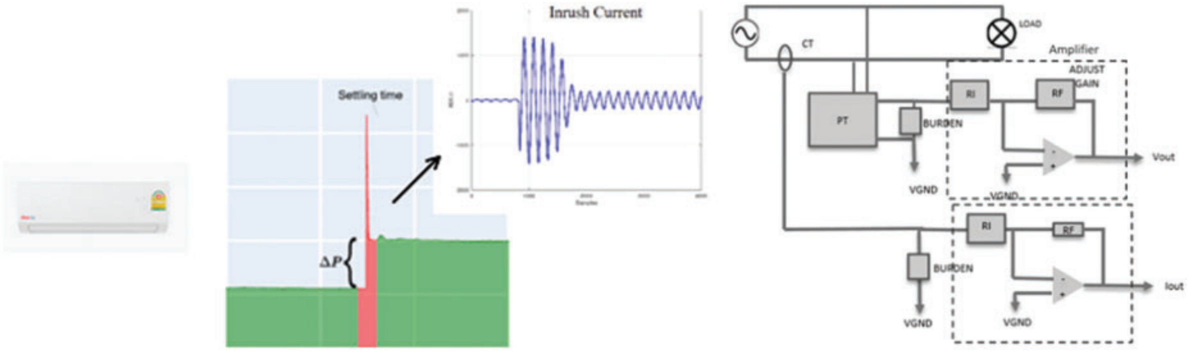
- ON/OFF state of appliances [17,18]: It refers to binary operation status, e.g., lamp by their switches, with short operational time. It is barely feasible for differentiation.
- Multi-state appliances [19]: The operational status of appliances changed with usage, e.g., the operation of kettles changes with adjusted values. It is not easy to understand signatures in any form of variables, and therefore, the analysis mainly relies on observation of their forms for a while to ensure their clarity.
- Variable power appliances [20]: The operational status of appliances changes continuously, e.g., the energy consumption of washing machines varies during washing/rinsing/steam. Users have to understand properties and observe the operation in the long run.
- Constant consumption appliances [10,21]: Appliances operate all the time, e.g., refrigerators or air-conditioners operate periodically. Besides, there are studies to change signatures of appliances by transforming data of operation (wave, voltage, and current) into images [13,22], which describes VI-trajectory with CNN, tested by PLAID and WHITED datasets. The [23] describes VI-trajectory with SNN, tested by PLAID, WHITED, and others [24,25].

The five appliances are set up to be five classes in this experiment which are labeled as Air condition1 (Class1), Air condition2 (Class2), Lamp (Class3), Microwave (Class4), and Pump (Class5). The characteristic of these appliances is shown in Tab. 1. This research focuses on improving the NILM system, and therefore these appliances are employed in different categories.

Table 1: Household appliances

Household appliance	Description	Category
Air condition (Class1)	7,033 Watt, 50 Hz, 7.5 Amp, 220 V(AC)	4
Air condition (Class2)	3,516 Watt, 50 Hz, 3.4 Amp, 220 V(AC)	4
Lamp (Class3)	28 Watt, 50 Hz, 0.431 Amp, 220 V(AC)	1
Microwave (Class4)	800 Watt, 50 Hz, 4.8 Amp, 220 V(AC)	2
Pump (Class5)	150 Watt, 50 Hz, 6.5 Amp, 220 V(AC)	3

When using appliances as load, the inrush current has occurred at the start point, which has different patterns depending on the capacitance and inductance of load power. This inrush current is a transient signal which is a signature of appliances. This research focuses on taking advantage of these patterns to analyze the appliance's operation in NILM. The example of inrush current is shown in Fig. 2a. The inrush current as the transient pattern is then collected using FPGA as data acquisition.



(a) (b)

Figure 2: Signal conditioning circuit

3.2 *Data Acquisition with FPGA*

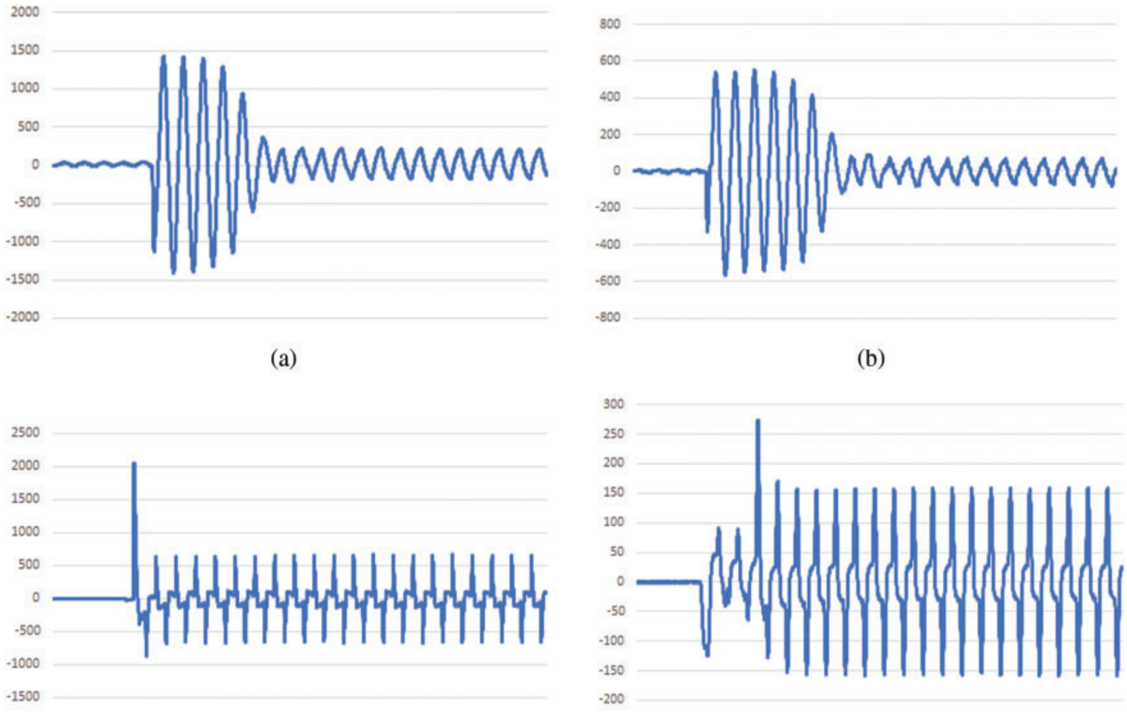
According to the proposed research focus to exploit the transient signal into the CNN classification method, this research designs the signal conditioning circuit to operate with FPGA. The signal conditioning circuit was designed Fig. 2b. It was used to measure the voltage values of the using appliances, passing through potential transformer and current transformer. Also, the signal conditioning circuit is designed to enable amplifiers and reduce noise. It consisted of two amplifier circuits to amplifier voltage signal and current signal. The ratio of an amplifier is 0.1–10 times. In the circuit, the R1 is fixed, but the Rf is adjustable between 1k–100 kOhms. The output of the circuit is Vout and Iout that are passed through the FPGA.

The 10M50DAF484C6GES FPGA chip is employed to correct data of the using appliances, which the algorithm of the FPGA is shown in Tab. 2. The Vout and Iout got into FPGA passed the smart analog (SMA) input (ADC1/2), which operated to read and check the data of the using appliances (STEP: 1 and STEP: 2). Check the correct operation of the switch (STEP: 3 and STEP: 4). The FPGA read the Vout and Iout at the ten μs sampling rate (STEP: 5 and STEP: 6). Then those data are passed through the computer via universal asynchronous receiver and transmitter (UART) port (STEP: 7). Save data at memory (STEP: 8), and the data converted to be image data using the kurtogram technique in the next step.

Table 2: FPGA algorithm for direct-attached storage (raw data)

```
INPUT: Digital signal data from signal conditioning circuit
STEP 1: Initialize GPIO
STEP 2 Config ADC1 and Config ADC2
STEP 3: IF SW start press ADC1; NO, Return STEP3
STEP 4: YES, Turn On load
STEP 5: GET ADC1 and GET ADC2 with wait 10 us
STEP 6: Maximum sampling?. 4,000 sampling; NO, Return STEP5
STEP 7: YES, Dumping voltage data via UART
STEP 8: Save voltage data at memory; Turn off the load and return STEP3
STEP 9: End
```

The inrush currents, which are sampled using FPGA, are shown in Fig. 3. It takes only 500 mS (25 cycles), which is a time advantage compared to the previous work 15 to be the input pattern of the kurtogram techniques. Moreover, it significantly shows the different characteristics, another advantage for classification when it is transformed into image formant by kurtogram technique.



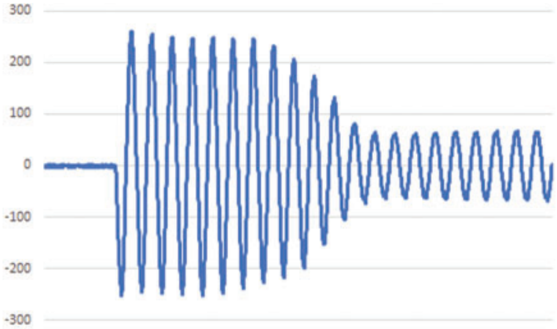
(a)

(b)

(c)

(d)

Figure 3: (Continued)



(e)

Figure 3: Inrush Current of each appliances; (a) Air condition (Class 1), (b) Air condition (Class2), (c) Lamp (Class 3), (d) Microwave (Class 4) and pump (Class 5)

3.3 Spectral Kurtosis and Kurtogram

In this section, a systematic approach is explained to transforming voltage signatures of appliances into kurtogram, and then CNN components for appliance classification. The Spectrum Kurtogram (SK) is used to descript the gaussian distribution statistic [50]. The SK can be expressed in Eqs. (1), to (3). The SK function is directly related to impulse response in the time domain.

$$a\,(n) = \int_{-\alpha}^{+\alpha} H\,(n,f)\, e^{j2\pi fn} dx\,(f) \tag{1}$$

where $H(n,f)$ is the complex envelope of in frequency domain is the component of the relation of SK with the conditionally non-stationary signal, which varies in space-time [52].

$$K_k\,(f) = \frac{\langle \|H\,(n,f)\,\|^4\rangle}{\langle \|H\,(n,f)\,\|^2\rangle^2} - 2 \tag{2}$$

where $K_k(f)$ is the SK which the conditionally non-stationary is a constant status in each frequency and fix the gaussian noise as zero. When the gaussian noise is not zero, it can be written as:

$$K_k + \phi\,(f) = \frac{K_k\,(f)}{[1 + \rho\,(f)]^2} \tag{3}$$

where $\phi(f)$ is the gaussian noise added in SK; $\rho(f)$ is the ratio of noise in using an appliance.

The inrush current of the air condition (Class1) is shown in Fig. 4a, which transfers to be image using the kurtogram technique based on Eqs. (1), to (3), as shown in Fig. 4b.

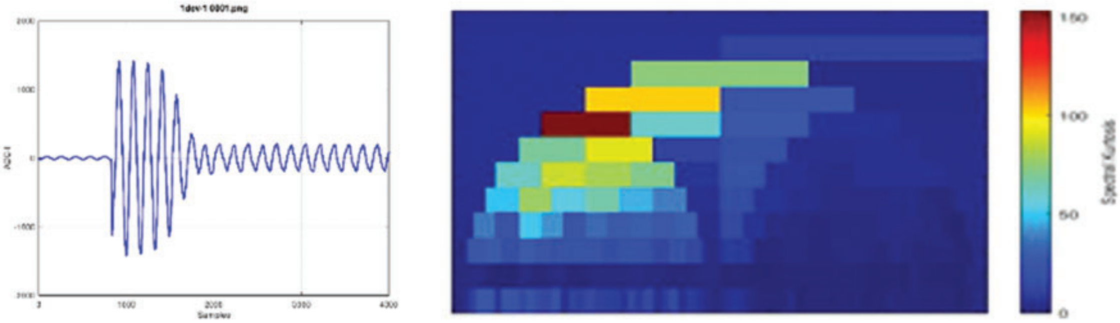
In the experiment, as identified in Fig. 4., the appliances can be randomly operated, which one or more than one appliance might operate simultaneously. Therefore, the inrush current or transient signal is divided into four conditions.

- Condition 1: When only one appliance activates, the transient signal is collected and then transferred to be image data to be data1, data 2, data 3 data 4, and data 5, respectively. These data are set to be Dataset 1.
- Condition 2: When two appliances activate, which classified to be five cases: class 1 + 2, class 1 + 5, class 2 + 5, class 3 + 4, and class 4 + 5, the transient signals are collected and then

transferred to be image data to be data12, data 15, data 25 data 34, and data 45, respectively. These data are set to be Dataset 2.

- Condition 3: When three appliances activate, which classified to be five cases: class 1 + 2+4, class 1 + 3+5, class 1 + 4+5, class 2 + 3+4, and class 3 + 4+5, the transient signals are collected and then transferred to be image data to be data124, data 135, data 145 data 234, and data 345, respectively. These data are set to be Dataset 3.
- Condition 4: When four appliances activate, which classified to be five cases: class 1 + 2+3 + 4, class 1 + 2+3 + 5, class 1 + 2+4 + 5, class 1 + 3+4 + 5, and class 2 + 3+4 + 5, the transient signals are collected and then transferred to be image data to be data1234, data 1235, data 1245 data 1345, and data 2345, respectively. These data are set to be Dataset 4.

The example of each dataset as an image pattern is shown in Fig. 5, which each image size is 100 × 100 pixels. Each dataset is contained 1,000 transient patterns, which separates to be training set and testing set as 80% and 20%, respectively, for CNN.



(a)

(b)

Figure 4: The inrush current of air condition (Class1) in kurtogram image; (a) Example inrush current data1 of air condition (Class1), (b) Kurtogram image of inrush current

3.4 CNN

The CNN is applied for classifying and recognizing the object [48,49] and other applications [13–15,21,28–32,35–38]. The CNN model of the proposed method is based on the VGGNet [49], which reduces the size to reduce the training time and faster operation. There are many ways to improve CNN classification accuracy, mainly based on the kernels convolutional. This research takes the image dataset to convolution with kernel as Eq. (4).

$$A * B\,(x, y) = \sum\nolimits_{ij} A\,(i, j)\, B\,(x + i, y + j) \tag{4}$$

when A is a 2D image, and K is the kernel convolution function.

The featured image, which passed through the feature extraction in each layer, as mentioned in Fig. 1., is calculated in ReLu activation function as shown in Eq. (5).

$$Y_j\,(n) = max(0, n) \tag{5}$$

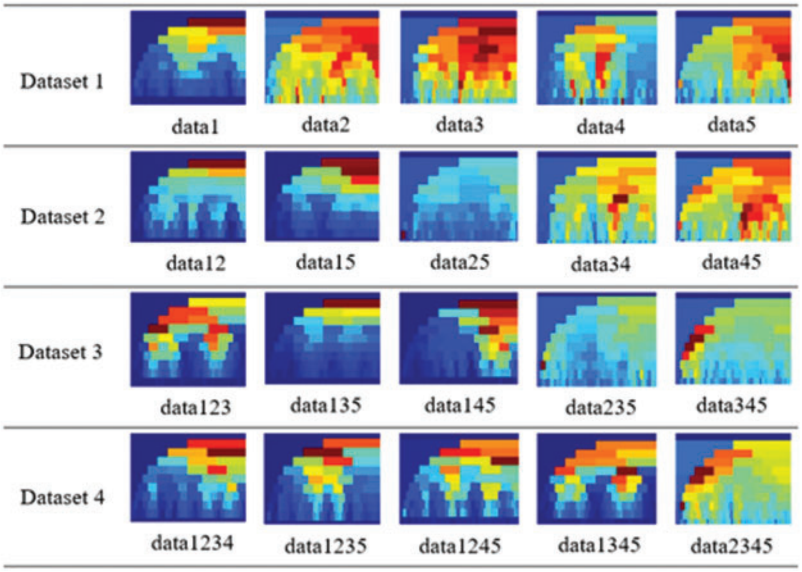


Figure 5: Kurtogram images of each dataset

Then the feature regularization operated with the DropBlock (DB) [45,46] for reducing the network overfitting according to the small dataset as shown in Eq. (6).

$$\gamma = \frac{1-KP}{BZ^2}\frac{FS^2}{(FS-BZ)^2} \tag{6}$$

when BZ is the size of the block to be dropped, KP is the keeping a unit in dropout.

The ANN structure is shown the Eq. (7).

$$V_j(n) = \sum\nolimits_{k=1}^{z}(W_{jk}X_k + b_j) \tag{7}$$

when $x = \{x_1, x_2, \ldots, x_n\}$ is the input, $W_{ji} = \{\, w_1, w_2, \ldots, w_n \}$ is the weight vector, (b_j) is the bias.

The z is the neuron number which the output has calculated the probability with the softmax function as shown in Eq. (8).

$$soft\max = \frac{exp(z_i)}{\sum_k exp(z_k)} \tag{8}$$

3.5 CNN Parameters

In the experiment, ten models of CNN are tested in the proposed method. It consists of three models of ImageNet CNN, two models of custom CNN, and five developing CNN models, namely CNN-A, CNN-B, CNN-C, CNN-D, and CNN-E. The AlexNet [48], VGG16 [49], and VGG19 [49], are chosen to use as ImageNet, which wildly used in image classification. It is applied to perform for NILM in this research. However, it provides high accuracy but takes too much time for NILM processing [13,53,54]. Another two models of custom CNN [55,56] were performed for NILM, and it reduced the model size to make a quick calculation. The five developing CNN models consist of 1) CNN-A modified from VGG11 [49], 2) CNN-B modified from VGG11which increases the kernel number and adapts the network to be U-Net [57]. 3) CNN-C modified from VGG13 [49], 4) CNN-D modified from VGG13, which increases the kernel, and 5) CNN-E modified from VGG16 [49]. The architecture of five developing CNN models is shown in Tab. 3.

Table 3: Configuration of difference method structure

Section (Block)	CNN-A	CNN-B	CNN-C	CNN-D	CNN-E
ConV Block1	ConV(32,3,3)(2L); AcT: “ReLu”; PaD: “Same” (all: 2 L); MaxP (2, 2); St: (2,2)	ConV(512,5,5)(3L); AcT: “ReLu”; PaD: “Same”; (all: 3 L); MaxP (2, 2); St: (2, 2); DB: (5, 0.70)	ConV(32,3,3)(1L)& ConV(64,3,3)(1L); (AcT: “ReLu”, PaD: “Same”, (all: 2 L)); MaxP (2, 2), St: (2,2)	ConV(64,3,3)(3L); (AcT: “ReLu”, PaD: “Same”, (all: 3 L)); MaxP (2, 2), St: (2,2)	ConV(128,5,5)(3L); (AcT: “ReLu”, PaD: “Same”, (all: 3 L)); MaxP (2, 2), St: (2,2)
ConV Block2	ConV(64,5,5)(2L); AcT: “ReLu”; PaD: “Same” (all: 2 L); MaxP (2, 2); St: (2,2)	ConV(64,5,5)(3L); AcT: “ReLu”; PaD: “Same”; (all: 3 L); MaxP (2, 2); St: (2,2); DB: (5, 0.70)	ConV(32,3,3)(1L)& ConV(64,3,3)(1L)& ConV(128,3,3)(1L); (AcT: “ReLu”, PaD: “Same”, (all: 3L)); MaxP (2, 2), St: (2,2)	ConV(64,5,5)(2L)& ConV(128,3,3)(1L); (AcT: “ReLu”, PaD: “Same”, (3 L)); MaxP (2, 2), St: (2,2)	ConV(128,3,3)(3L); AcT: “ReLu”; PaD: “Same” (all: 3 L); MaxP (2, 2); St: (2,2)
ConV Block3	ConV(32,3,3)(2L); AcT: “ReLu”; PaD: “Same” (all: 2 L); MaxP (2, 2); St: (2,2)	ConV(32,3,3)(3L); AcT: “ReLu”; PaD: “Same” (all: 3 L); MaxP (2, 2); St: (2,2); DB: (5, 0.70)	ConV(32,3,3)(1L)& ConV(64,3,3)(1L)& ConV(128,3,3)(1L); (AcT: “ReLu”, PaD: “Same”, (all: 3L)); MaxP (2, 2), St: (2,2)	ConV(32,3,3)(1L)& ConV(64,3,3)(2L); (AcT: “ReLu”, PaD: “Same”, (3 L)); MaxP (2, 2), St: (2,2)	ConV(256,5,5)(3L); AcT: “ReLu”; PaD: “Same” (all: 3 L); MaxP (2, 2); St: (2,2)
ConV Block4	ConV(64,5,5)(2L); AcT: “ReLu”; PaD: “Same” (all: 2 L); MaxP (2, 2); St: (2,2)	ConV(64,5,5)(3L); AcT: “ReLu”; PaD: “Same” (all: 3 L); MaxP (2, 2); St: (2,2); DB: (5, 0.70)	ConV(128,5,5)(1L)& ConV(256,5,5) (1L)& ConV(512,5,5) (1L); (AcT: “ReLu”, PaD: “Same”, (all: 3L)); MaxP (2, 2), St: (2,2); DB: (2, 0.90)	ConV(128,5,5)(2L)& ConV(256,5,5) (1L); (AcT: “ReLu”, PaD: “Same”, (all: 3 L)); MaxP (2, 2), St: (2,2)	ConV(512,5,5)(3L); AcT: “ReLu”; PaD: “Same” (all: 3 L); MaxP (2, 2); St: (2,2)
ConV Block5	ConV(64,5,5)(2L); AcT: “ReLu”; PaD: “Same” (all: 2 L); MaxP (2, 2); St: (2,2); DB: (2, 0.90)	ConV(128,5,5)(3L); AcT: “ReLu”; PaD: “Same” (all: 3 L); MaxP (2, 2); St: (2,2); DB: (5, 0.70)	ConV(512,5,5)(3L); AcT: “ReLu”; PaD: “Same” (all: 3L); MaxP (2, 2); St: (2,2); DB: (2, 0.90)	ConV(256,5,5)(3L); AcT: “ReLu”; PaD: “Same” (all: 3 L); MaxP (2, 2); St: (2,2); DB: (2, 0.90)	ConV(1024,3,3)(3L); AcT: “ReLu”; PaD: “Same” (all: 3 L); MaxP (2, 2); St: (2,2)
Classification Block	FC1: 500; (ReLu) FC2: 500: (ReLu) SoftMax	FC1: 1024; (ReLu) FC2: 1024: (ReLu) SoftMax	FC1: 1024; (ReLu) FC2: 1024: (ReLu) SoftMax	FC1: 1024; (ReLu) FC2: 1024: (ReLu) SoftMax	FC1: 1024; (ReLu) FC2: 1024: (ReLu) SoftMax

#Note: ConV: Convolutional Layer; L: Layers; AcT: Activation function; PaD: Padding; MaxP: MaxPooling; DB: Dropblock; St: Strid.

The CNN-A to CNN-E was tested with the image sized 100 × 100 × 3 pixel and model trained with the 100 epoch. Before the training, the learning rate was adjusted to 0.0001, with the Adam optimized was set at 0.9 and 0.0009. Cross-entropy loss measures the performance of a classification model. This experiment was performed on Windows 10, RAM 32 GB, the graphics card model the RTX 2070 with GPU memory 8 GB to accelerate the calculation speed. The researcher applied the library of Tensorflow and Numpy in Python 3.6.

In training, the parameters of 10 CNN models are shown in Tab. 4, in which AlexNet [13], VGG16 [49], VGG19 [49], CNN [55,56] are set the parameter as the previous works.

Table 4: Training parameter

Algorithms for NILM techniques	Images Size	Learning rate	Epoch	Batch size
AlexNet [13]	32 × 32 × 3	0.0001	100	24
VGG16 [49]	224 × 224 × 3	0.00001	100	24
VGG19 [49]	224 × 224 × 3	0.00001	100	24
CNN [55]	50 × 50 × 3	0.0001	100	24
CNN [56]	50 ×50 × 3	0.001	100	24
CNN-A (VGG11)	100 × 100 × 3	0.0001	100	24
CNN-B (VGG11)	100 × 100 × 3	0.0001	100	24
CNN-C (VGG13)	100 × 100 × 3	0.0001	100	24
CNN-D (VGG13)	100 × 100 × 3	0.0001	100	24
CNN-E (VGG16)	100 × 100 × 3	0.0001	100	24

3.6 Evaluation

Accuracy (ACC) is a commonly used evaluation metric, and we use the accuracy to evaluate the model's overall performance, calculated as the Eq. (9). Precision as the learning curve of the model, calculated as the Eq. (10). Recall and F1-score are performance metrics that apply to appliance classification, calculated as the Eqs. (11) and (12). Mean square error (MSE), and root mean square error (RMSE) are used to evaluate the model in regression analysis, calculated as the Eqs. (13), and (14).

$$\text{Accuracy} = (\text{TP} + \text{TN}) \,/\, (\text{TP} + \text{TN} + \text{FP} + \text{FN}) \tag{9}$$

$$\text{Precision} = \text{TP}/\,(\text{TP} + \text{FP}) \tag{10}$$

$$\text{Recall} = \text{TP}/\,(\text{TP} + \text{FN}) \tag{11}$$

$$\text{F1} - \text{score} = 2 \times (\text{Recall} \times \text{Precision}) \,/\, (\text{Recall} + \text{Precision}) \tag{12}$$

$$\text{MSE} = \frac{1}{2}\sum\nolimits_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n} \tag{13}$$

$$\text{RMSE} = \sqrt{MSE} \tag{14}$$

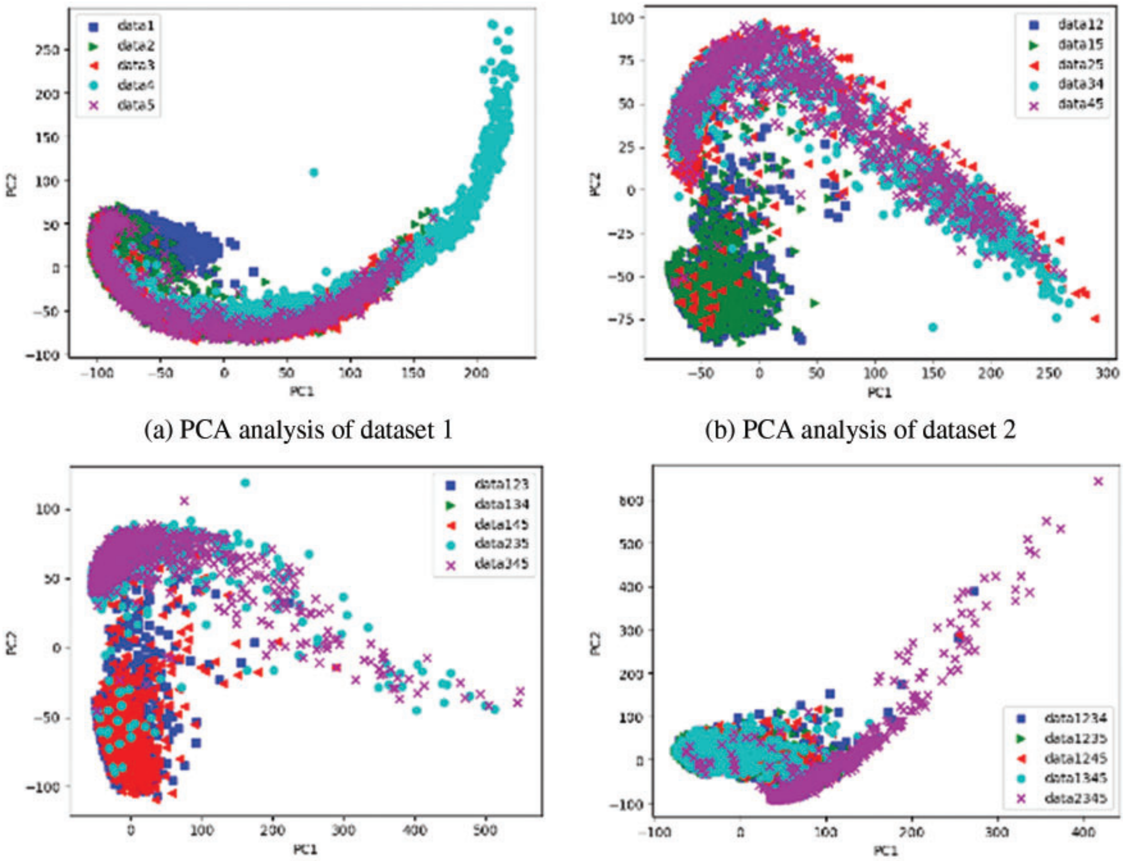
when TP is True Positive; TN is True Negative; FP is False Positive; FN is False Negative.

4 Result and Discussion

All the result is divided into three parts; 1) the variance on kurtogram data set analysis, 2) the result of the testing model, and 3) the result and accuracy of the CNN model.

4.1 The Variance on Kurtogram Dataset Analysis

The principal component analysis (PCA) is used for data distributed analysis of the dataset, as shown in Fig. 6.



(a) PCA analysis of dataset 1

(b) PCA analysis of dataset 2

(c) PCA analysis of dataset 3

(d) PCA analysis of dataset 4

Figure 6: PCA analysis of the dataset

Fig. 6a, isolates the dataset 1 distribution, which only one appliance activates. It was found that the data distribution was overlapping, and there was no clear grouping. Therefore, the data from the entire dataset was considered suitable for the CNN model testing as the data was not grouped previously. Figs. 6b and 6c, show that clustering occurs for some data sets such as data12 or data15. However, they still overlap with other data, and in Fig. 6d, four appliances are switched on simultaneously. They all overlap, meaning they are not grouped. So, it is a dataset suitable for CNN analysis.

4.2 Result and Accuracy of CNN Model

The load recognition accuracy values obtained from 10 CNN models are shown in Tab. 5. The highest accuracy in each dataset is the CNN-E (VGG16), which provides an accuracy of 99.96% for dataset1. The CNN-E (VGG16) has an accuracy of 99.09% for dataset2. The CNN-E (VGG16) has an accuracy of 99.55% for dataset3, and the CNN-D (VGG13) has an accuracy of 98.38% for dataset4. The highest accuracy compared with the best of each accuracy dataset is the CNN-E (VGG16) which gives an accuracy of 99.96%. Therefore, it is suitable to apply for NILM with the kurtogram technique.

Table 5: Accuracy comparing deep CNN for appliance classifiers

CNN model	Dataset 1 (ACC %)	Dataset 2 (ACC %)	Dataset 3 (ACC %)	Dataset 4 (ACC %)	Average (ACC %)
AlexNet [13]	99.34 ± 0.18	93.21 ± 1.26	86.83 ± 0.49	80.22 ± 1.10	89.90
VGG16 [49]	99.30 ± 0.11	97.89 ± 0.15	91.53 ± 0.46	89.94 ± 1.12	94.67
VGG19 [49]	99.15 ± 0.12	95.04 ± 0.20	87.17 ± 1.34	82.08 ± 1.44	90.86
CNN [55]	95.77 ± 1.03	83.96 ± 1.59	77.20 ± 1.02	74.03 ± 1.58	82.74
CNN [56]	99.28 ± 0.13	98.34 ± 0.22	97.10 ± 0.64	97.76 ± 1.02	98.12
CNN-A (VGG11)	99.30 ± 0.13	96.52 ± 1.11	95.65 ± 0.33	93.59 ± 1.05	96.27
CNN-B (VGG11)	97.43 ± 0.49	90.42 ± 1.27	85.59 ± 1.03	69.57 ± 1.53	85.75
CNN-C (VGG13)	99.26 ± 0.15	96.87 ± 1.28	99.05 ± 0.16	97.85 ± 0.33	98.26
CNN-D (VGG13)	99.69 ± 0.21	98.31 ± 0.32	99.32 ± 0.06	98.38 ± 0.52	98.93
CNN-E (VGG16)	99.96 ± 0.01	99.09 ± 0.16	99.55 ± 0.11	97.65 ± 0.36	**99.06**

The training time of each model is shown in Tab. 6. It was noticed that the developing network CNN-E (VGG16) provided a training time of 44.39 min. It is the longest training time compared to all ten models; however, it provides the highest accuracy. At this point, accuracy values are more important than the training time because of the system's efficiency. The training time can make it faster if the resources are upgraded. Thus, developing network CNN-E (VGG16) is suitable for NILM with the kurtogram technique.

Table 6: Training time

CNN Model	Dataset 1 Time (m)	Dataset 2 Time (m)	Dataset 3 Time (m)	Dataset 4 Time (m)	Average Time (m)
AlexNet [13]	11.2 ± 1.56	11.2 ± 0.58	12.4 ± 1.15	12.4 ± 1.16	11.80
VGG16 [49]	32.7 ± 1.28	37.4 ± 1.44	41.4 ± 1.02	37.2 ± 1.41	37.18
VGG19 [49]	69.5 ± 1.32	98.4 ± 2.04	75.3 ± 2.36	58.5 ± 1.53	75.43
CNN [55]	4.08 ± 0.51	4.23 ± 0.30	4.04 ± 0.31	3.50 ± 0.26	3.96
CNN [56]	3.24 ± 0.11	3.49 ± 0.32	3.31 ± 0.11	3.40 ± 0.18	**3.36**
CNN-A (VGG11)	05.29 ± 0.21	05.35 ± 0.36	05.30 ± 0.32	05.12 ± 0.46	5.27
CNN-B (VGG11)	74.05 ± 1.26	21.16 ± 1.48	81.85 ± 1.44	39.53 ± 2.02	54.15
CNN-C (VGG13)	17.11 ± 1.15	17.12 ± 1.46	15.06 ± 0.55	25.05 ± 1.02	18.59

(Continued)

Table 6: Continued

CNN Model	Dataset 1 Time (m)	Dataset 2 Time (m)	Dataset 3 Time (m)	Dataset 4 Time (m)	Average Time (m)
CNN-D (VGG13)	14.51 ± 1.28	15.12 ± 1.51	13.45 ± 1.08	13.66 ± 1.49	14.19
CNN-E (VGG16)	45.83 ± 2.12	45.03 ± 1.53	41.41 ± 1.45	45.30 ± 1.16	44.39

4.3 Result of Classification

The F1-score and Recall are used to measure the classification of testing data of the five developing CNN models as CNN-A to CNN-E as shown in Tab. 7.

- In condition 1: the F1-score provides the best accuracy as 1.00, 0.96, 0.96, 0.99, and 0.96, and also the Recall offers 1.00, 0.99, 0.95, 0.99, and 0.96 for data1, data2, data3, data4, and data5, respectively. The best developing CNN model is the CNN-C which gives the highest average F1-score and average Recall as 0.970 and 0.970, respectively. At the same time, the CNN-B offers the lowest average F1-score and average Recall in data3.
- In condition 2: the F1-score provides the best accuracy as 0.96, 0.92, 0.92, 0.99 and 0.96, and also the Recall offers 0.96, 0.92, 0.92, 0.99, and 0.96 for data15, data25, data34, and data45, respectively. The best developing CNN model is the CNN-E which gives the highest average F1-score and average Recall as 0.942 and 0.942, respectively. At the same time, the CNN-B offers the lowest average F1-score and average Recall in appliance class 1.
- In condition 3: the F1-score provides the best accuracy as 0.60, 0.68, 0.65, 0.86 and 0.86, and also the Recall offers 0.57, 0.74, 0.68, 0.93, and 0.86 for data124, data135, data145, data 235and data345, respectively. The best developing CNN model is the CNN-C which gives the highest average F1-score and average Recall as 0.726 and 0.726, respectively. At the same time, the CNN-C offers the lowest average F1-score and average Recall in appliance classes 1 and 4.
- In condition 4: the F1-score provides the best accuracy as 0.53, 0.78, 0.55, 0.68 and 0.98, and also the Recall offers 0.61, 0.75, 0.59, 0.65, and 0.96 for data1234, data1245, data1345, data 1235 and data2345, respectively. The best developing CNN model is the CNN-D which gives the highest average F1-score and average Recall as 0.702 and 0.698, respectively. At the same time, the CNN-B offers the lowest average F1-score and average Recall in appliance classes 1 and 4.

It can be summary that the developing CNN-C, D, and E is suitable for this NILM experiment. On the opposite side, the CNN-B model is unsuitable for this NILM experiment. Moreover, when the appliance operates simultaneously, it reduces the efficiency of the proposed system. Also, the high-power load appliance, which offers the high transients signal, gives a noise to classify function.

Table 7: Performance comparison of appliance classifiers

Appliance	CNN-A		CNN-B		CNN-C		CNN-D		CNN-E	
(class)	F1-score	Recall	F1-score	Recall	F1-score	Recall	F1-score	Recall	F1-score	Recall
Dataset 1										
data1	0.99 ± 0.01	1.00 ± 0.00	0.99 ± 0.01	0.99 ± 0.01	1.00 ± 0.00	0.99 ± 0.01	0.99 ± 0.01	0.99 ± 0.01	0.99 ± 0.01	0.99 ± 0.01
data2	0.96 ± 0.01	0.99 ± 0.01	0.91 ± 0.03	0.96 ± 0.01	0.96 ± 0.02	0.99 ± 0.01	0.96 ± 0.01	0.98 ± 0.01	0.95 ± 0.04	0.99 ± 0.01
data3	0.95 ± 0.02	0.91 ± 0.03	0.87 ± 0.05	0.77 ± 0.02	0.96 ± 0.01	0.95 ± 0.02	0.95 ± 0.02	0.92 ± 0.01	0.95 ± 0.02	0.90 ± 0.02
data4	0.99 ± 0.01	0.99 ± 0.01	0.99 ± 0.01	0.99 ± 0.01	0.99 ± 0.01	0.99 ± 0.01	0.99 ± 0.01	0.99 ± 0.01	0.99 ± 0.01	0.99 ± 0.01
data5	0.93 ± 0.01	0.94 ± 0.02	0.86 ± 0.02	0.88 ± 0.02	0.94 ± 0.01	0.93 ± 0.01	0.95 ± 0.01	0.96 ± 0.01	0.96 ± 0.01	0.96 ± 0.04
Average	0.964	0.966	0.924	0.918	**0.970**	**0.970**	0.968	0.968	0.968	0.966
Dataset2										
data12	0.85 ± 0.06	0.85 ± 0.08	0.72 ± 0.02	0.74 ± 0.02	0.90 ± 0.04	0.92 ± 0.04	0.91 ± 0.03	0.93 ± 0.07	0.92 ± 0.01	0.96 ± 0.01
data15	0.85 ± 0.06	0.87 ± 0.05	0.72 ± 0.01	0.71 ± 0.03	0.92 ± 0.01	0.92 ± 0.07	0.90 ± 0.04	0.89 ± 0.02	0.92 ± 0.08	0.89 ± 0.02
data25	0.94 ± 0.04	0.92 ± 0.03	0.92 ± 0.01	0.86 ± 0.01	0.94 ± 0.01	0.90 ± 0.03	0.94 ± 0.05	0.92 ± 0.02	0.94 ± 0.02	0.92 ± 0.01
data34	0.98 ± 0.01	0.99 ± 0.04	0.92 ± 0.06	0.90 ± 0.05	0.96 ± 0.02	0.99 ± 0.01	0.97 ± 0.02	0.98 ± 0.01	0.98 ± 0.06	0.98 ± 0.01
data45	0.96 ± 0.03	0.95 ± 0.06	0.90 ± 0.05	0.95 ± 0.02	0.96 ± 0.05	0.96 ± 0.02	0.95 ± 0.01	0.96 ± 0.05	0.95 ± 0.02	0.96 ± 0.05
Average	0.916	0.916	0.836	0.832	0.936	0.938	0.934	0.936	**0.942**	**0.942**
Dataset3										
data124	0.55 ± 0.01	0.55 ± 0.01	0.35 ± 0.03	0.31 ± 0.05	0.60 ± 0.02	0.57 ± 0.04	0.58 ± 0.02	0.55 ± 0.01	0.58 ± 0.08	0.52 ± 0.01
data135	0.60 ± 0.05	0.55 ± 0.04	0.46 ± 0.06	0.38 ± 0.08	0.68 ± 0.02	0.73 ± 0.02	0.62 ± 0.04	0.64 ± 0.01	0.67 ± 0.03	0.74 ± 0.01
data145	0.63 ± 0.02	0.67 ± 0.04	0.56 ± 0.01	0.68 ± 0.05	0.65 ± 0.03	0.63 ± 0.01	0.61 ± 0.01	0.60 ± 0.02	0.63 ± 0.01	0.60 ± 0.03
data235	0.83 ± 0.02	0.85 ± 0.03	0.79 ± 0.02	0.75 ± 0.01	0.86 ± 0.01	0.92 ± 0.05	0.82 ± 0.05	0.87 ± 0.04	0.86 ± 0.02	0.93 ± 0.01
data345	0.82 ± 0.01	0.79 ± 0.02	0.83 ± 0.02	0.86 ± 0.01	0.84 ± 0.01	0.78 ± 0.02	0.84 ± 0.02	0.80 ± 0.02	0.85 ± 0.02	0.77 ± 0.06
Average	0.686	0.682	0.598	0.596	**0.726**	**0.726**	0.694	0.692	0.718	0.712
Dataset4										
data1234	0.48 ± 0.09	0.52 ± 0.03	0.24 ± 0.07	0.32 ± 0.08	0.52 ± 0.04	0.61 ± 0.03	0.53 ± 0.05	0.56 ± 0.01	0.50 ± 0.08	0.55 ± 0.01
data1235	0.75 ± 0.02	0.71 ± 0.02	0.45 ± 0.02	0.33 ± 0.05	0.77 ± 0.08	0.68 ± 0.01	0.78 ± 0.09	0.75 ± 0.03	0.75 ± 0.03	0.70 ± 0.03
data1245	0.53 ± 0.04	0.53 ± 0.05	0.28 ± 0.02	0.21 ± 0.05	0.55 ± 0.02	0.59 ± 0.05	0.55 ± 0.06	0.57 ± 0.03	0.54 ± 0.01	0.58 ± 0.01
data1345	0.65 ± 0.03	0.64 ± 0.01	0.36 ± 0.05	0.29 ± 0.04	0.68 ± 0.02	0.64 ± 0.03	0.67 ± 0.03	0.65 ± 0.01	0.66 ± 0.04	0.62 ± 0.01
data2345	0.98 ± 0.01	0.95 ± 0.01	0.96 ± 0.01	0.93 ± 0.02	0.98 ± 0.01	0.96 ± 0.02	0.98 ± 0.01	0.96 ± 0.01	0.98 ± 0.01	0.95 ± 0.02
Average	0.678	0.670	0.458	0.416	0.700	0.696	**0.702**	**0.698**	0.686	0.680

5 Conclusion

NILM mainly monitors energy consumption by using computational intelligence and sensing technology. The proposed NILM system involves three main processes: data collection, data processing, and classification. This system identifies the appliance using the inrush current pattern with the modified CNN models. The advantage of the high sampling rate of FPGA is employed to sample the inrush current of five appliances, and the kurtogram technique transforms those inrush current to be image patterns. Then the ten CNN models, of which five are developed in this research, are operated to train and test those patterns. It can be noticed that the developing CNN model, which is based on VGG16, provided the highest average accuracy as 99.06%. The proposed system is suitable to apply to the NILM system.

Acknowledgement: The authors would like to express sincere gratitude to the Signal Processing Research Laboratory, the Faculty of Engineering, Rajamangala University of Technology Thanyaburi, for insight and expertise.

Funding Statement: This research has received funding support from the NSRF via the Program Management Unit for Human Resources & Institutional Development, Research and Innovation (Grant number BO4G640045). Also, this research is supported by the National Research Council of Thailand (NRCT). NRISS No. 144276, 2589514 (FFB65E0712) and 2589488 (FFB65E0713).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] P. Franco, J. M. Martínez, Y. C. Kim and M. A. Ahmed, "IoT based approach for load monitoring and activity recognition in smart homes," *IEEE Access*, vol. 9, pp. 45325–45339, 2021.

[2] A. Kadechkar, J. Riba, M. M. Eguilaz and J. Sanllehí, "Real-time wireless, contactless, and coreless monitoring of the current distribution in substation conductors for fault diagnosis," *IEEE Sensors Journal*, vol. 19, no. 5, pp. 1693–1700, 2019.

[3] X. Kong, S. Zhu, X. Huo, S. Li, Y. Li *et al.,* "A household energy efficiency index assessment method based on non-intrusive load monitoring data," *Applied Sciences*, vol. 10, no. 11, pp. 1–21, 2020.

[4] H. H. Chang and N. V. Linh, "Statistical feature extraction for fault locations in nonintrusive fault detection of low voltage distribution systems," *Energies*, vol. 10, no. 5, pp. 1–19, 2017.

[5] A. U. Rehman, T. T. Lie, B. Vallès and S. R. Tito, "Non-intrusive load monitoring of residential water-heating circuit using ensemble machine learning techniques," *Inventions*, vol. 5, no. 4, pp. 1–20, 2020.

[6] L. Massidda, M. Marrocu and S. Manca, "Non-intrusive load disaggregation by convolutional neural network and multilabel classification," *Applied Sciences*, vol. 10, no. 4, pp. 1–17, 2020.

[7] B. Cannas, S. Carcangiu, D. Carta, A. Fanni and C. Muscas, "Selection of features based on electric power quantities for non-intrusive load monitoring," *Applied Sciences*, vol. 11, no. 2, pp. 1–14, 2021.

[8] T. K. Nguyen, E. Dekneuvel, G. Jacquemod, B. Nicolle, O. Zammit *et al.,* "Development of a real-time non-intrusive appliance load monitoring system: An application level model," *International Journal of Electrical Power & Energy Systems*, vol. 90, pp. 168–180, 2017.

[9] F. D. Garcia, W. A. Souza, I. S. Diniz and F. P. Marafão, "NILM-Based approach for energy efficiency assessment of household appliances," *Energy Informatics*, vol. 3, no. 1, pp. 10, 2020.

[10] M. Qureshi, C. Ghiaus and N. Ahmad, "A blind event-based learning algorithm for non-intrusive load disaggregation," *International Journal of Electrical Power & Energy Systems*, vol. 129, pp. 106834, 2021.

[11] H. K. Iqbal, F. H. Malik, A. Muhammad, M. A. Qureshi, M. N. Abbasi *et al.,* "A critical review of state-of-the-art non-intrusive load monitoring datasets," *Electric Power Systems Research*, vol. 192, pp. 106921, 2021.

[12] L. De Baets, J. Ruyssinck, C. Develder, T. Dhaene, D. Deschrijver *et al.,* "Appliance classification using vi trajectories and convolutional neural networks," *Energy and Buildings*, vol. 158, pp. 32–36, 2018.

[13] Y. Liu, X. Wang and W. You, "Non-intrusive load monitoring by voltage–current trajectory enabled transfer learning," *IEEE Transactions on Smart Grid*, vol. 10, no. 5, pp. 5609–5619, 2019.

[14] X. Wu, X. Han, L. Liu and B. Qi, "A load identification algorithm of frequency domain filtering under current underdetermined separation," *IEEE Access*, vol. 6, pp. 37094–37107, 2018.

[15] J. Song, Y. Lee and E. Hwang, "Time–frequency mask estimation based on deep neural network for flexible load disaggregation in buildings," *IEEE Transactions on Smart Grid*, vol. 12, no. 4, pp. 3242–3251, 2021.

[16] B. Zhao, K. He, L. Stankovic and V. Stankovic, "Improving event-based non-intrusive load monitoring using graph signal processing," *IEEE Access*, vol. 6, pp. 53944–53959, 2018.

[17] Y. Liu, X. Wang, L. Zhao and Y. Liu, “Admittance-based load signature construction for non-intrusive appliance load monitoring,” *Energy and Buildings*, vol. 171, pp. 209–219, 2018.

[18] S. Ghosh and D. Chatterjee, “Non-intrusive identification of harmonic polluting loads in a smart residential system,” *Sustainable Energy, Grids and Networks*, vol. 26, pp. 100446, 2021.

[19] K. Chen, Y. Zhang, Q. Wang, J. Hu, H. Fan, H. Jinliang *et al.,* “Scale- and context-aware convolutional non-intrusive load monitoring,” *IEEE Transactions on Power Systems*, vol. 35, no. 3, pp. 2362–2373, 2019.

[20] K. Basu, V. Debusschere, A. Douzal-Chouakria and S. Bacha, “Time series distance-based methods for non-intrusive load monitoring in residential buildings,” *Energy and Buildings*, vol. 96, pp. 109–117, 2015.

[21] Z. Zhou, Y. Xiang, H. Xu, Z. Yi, D. Shi, Z. Wang *et al.,* “A novel transfer learning-based intelligent nonintrusive load-monitoring with limited measurements,” *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–8, 2021.

[22] L. Du, D. He, R. G. Harley and T. G. Habetler, “Electric load classification by binary voltage–current trajectory mapping,” *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 358–365, 2016.

[23] L. De Baets, C. Develder, T. Dhaene and D. Deschrijver, “Detection of unidentified appliances in non-intrusive load monitoring using siamese neural networks,” *International Journal of Electrical Power & Energy Systems*, vol. 104, pp. 645–653, 2019.

[24] A. L. Wang, B. X. Chen, C. G. Wang and D. Hua, “Non-intrusive load monitoring algorithm based on features of v–i trajectory,” *Electric Power Systems Research*, vol. 157, pp. 134–144, 2018.

[25] D. Baptista, S. Mostafa, L. Pereira, L. Sousa, F. Morgado-Dias *et al.,* “Implementation strategy of convolution neural networks on field programmable gate arrays for appliance classification using the voltage and current (v-i) trajectory,” *Energies*, vol. 11, pp. 2460, 2018.

[26] J. M. Vann, T. Karnowski and A. L. Anderson, “Classification of unintended radiated emissions in a multi-device environment,” *IEEE Transactions on Smart Grid*, vol. 10, no. 5, pp. 5506–5513, 2019.

[27] D. Yang, X. Gao, L. Kong, Y. Pang, B. Zhou *et al.,* “An event-driven convolutional neural architecture for non-intrusive load monitoring of residential appliance,” *IEEE Transactions on Consumer Electronics*, vol. 66, no. 2, pp. 173–182, 2020.

[28] K. Li, B. Yin, Z. Du and Y. Sun, “A Non-intrusive load identification model based on time-frequency features fusion,” *IEEE Access*, vol. 9, pp. 1376–1387, 2021.

[29] B. Yin, L. Zhao, X. Huang, Y. Zhang, Z. Du *et al.,* “Research on non-intrusive unknown load identification technology based on deep learning,” *International Journal of Electrical Power & Energy Systems*, vol. 131, pp. 107016, 2021.

[30] F. Ciancetta, G. Bucci, E. Fiorucci, S. Mari and A. Fioravanti, “A new convolutional neural network-based system for nilm applications,” *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–12, 2021.

[31] E. Gomes and L. Pereira, “PB-NILM: Pinball guided deep non-intrusive load monitoring,” *IEEE Access*, vol. 8, pp. 48386–48398, 2020.

[32] M. Kaselimi, E. Protopapadakis, A. Voulodimos, N. Doulamis and A. Doulamis, “Multi-channel recurrent convolutional neural networks for energy disaggregation,” *IEEE Access*, vol. 7, pp. 81047–81056, 2019.

[33] M. Kaselimi, N. Doulamis, A. Voulodimos, E. Protopapadakis, A. Doulamis *et al.,* “Context aware energy disaggregation using adaptive bidirectional lstm models,” *IEEE Transactions on Smart Grid*, vol. 11, no. 4, pp. 3054–3067, 2020.

[34] J. Baek, M. W. Kanampiu and C. Kim, “A secure internet of things smart home network: Design and configuration,” *Applied Sciences*, vol. 11, no. 14, pp. 1–16, 2021.

[35] G. Cui, B. Liu, W. Luan and Y. Yu, “Estimation of target appliance electricity consumption using background filtering,” *IEEE Transactions on Smart Grid*, vol. 10, no. 6, pp. 5920–5929, 2019.

[36] M. Xia, W. A. Liu, K. Wang, X. Zhang, Y. Xu *et al.,* “Non-intrusive load disaggregation based on deep dilated residual network,” *Electric Power Systems Research*, vol. 170, pp. 277–285, 2019.

[37] M. Xia, W. A. Liu, Y. Xu, K. Wang, X. Zhang *et al.,* “Dilated residual attention network for load disaggregation,” *Neural Computing and Applications*, vol. 31, no. 12, pp. 8931–8953, 2019.

[38] Z. Jia, L. Yang, Z. Zhang, H. Liu, F. Kong *et al.,* "Sequence to point learning based on bidirectional dilated residual network for non-intrusive load monitoring," *International Journal of Electrical Power & Energy Systems*, vol. 129, pp. 106837, 2021.

[39] M. Dong, P. C. M. Meira, W. Xu and C. Y. Chung, "Non-intrusive signature extraction for major residential loads," *IEEE Transactions on Smart Grid*, vol. 4, no. 3, pp. 1421–1430, 2013.

[40] S. Ghosh, D. Manna, A. Chatterjee and D. Chatterjee, "Remote appliance load monitoring and identification in a modern residential system with smart meter data," *IEEE Sensors Journal*, vol. 21, no. 4, pp. 5082–5090, 2021.

[41] L. D. Baets, J. Ruyssinck, C. Develder, T. Dhaene, D. Eschrijver *et al.,* "On the Bayesian optimization and robustness of event detection methods in ilm," *Energy and Buildings*, vol. 145, pp. 57–66, 2017.

[42] N. Henao, K. Agbossou, S. Kelouwani, Y. Dubé, M. Fournier *et al.,* "Approach in nonintrusive type i load monitoring using subtractive clustering," *IEEE Transactions on Smart Grid*, vol. 8, no. 2, pp. 812–821, 2017.

[43] T. Hassan, F. Javed and N. Arshad, "An empirical investigation of v-i trajectory based load signatures for non-intrusive load monitoring," *IEEE Transactions on Smart Grid*, vol. 5, no. 2, pp. 870–878, 2014.

[44] J. Gao, E. C. Kara, S. Giri and M. Bergés, "A feasibility study of automated plug-load identification from high-frequency measurements," in *IEEE Global Conf. on Signal and Information Processing (GlobalSIP)*, Florida, USA, pp. 220–224, 2015.

[45] H. Altrabalsi, J. Liao, L. Stankovic and V. Stankovic, "A Low-complexity energy disaggregation method: Performance and robustness," in *IEEE Symp. on Computational Intelligence Applications in Smart Grid (CIASG)*, Florida, USA, pp. 1–8, 2014.

[46] M. Nguyen, S. Alshareef, A. Gilani and W. G. Morsi, "A novel feature extraction and classification algorithm based on power components using single-point monitoring for NILM," in *IEEE 28th Canadian Conf. on Electrical and Computer Engineering (CCECE)*, Nova, Scotia, pp. 37–40, 2015.

[47] K. Basu, A. Hably, V. Debusschere, S. Bacha, A. Ovalle *et al.,* "A comparative study of low sampling non intrusive load dis-aggregation," In *42nd Annual Conf. of the IEEE Industrial Electronics Society*, Florence, Italy, pp. 5137–5142, 2016.

[48] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. of the Int. Conf. on Learning Representations (ICLR 2015)*, San Diego, CA, USA, pp. 1409.1556, 2015.

[49] A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Int. Conf. on Neural Information Processing Systems*, Montréal CANADA, pp. 1097–1105, 2015.

[50] J. Antoni, "The spectral kurtosis: A useful tool for characterising non-stationary signals," *Mechanical Systems and Signal Processing*, vol. 20, no. 2, pp. 282–307, 2006.

[51] C. Xianglong, F. Fuzhou and Z. Bingzhi, "Weak fault feature extraction of rolling bearings based on an improved kurtogram," *Sensors*, vol. 16, no. 9, pp. 1482, 2016.

[52] S. Udmale and S. Singh, "A mechanical data analysis using kurtogram and extreme learning machine," *Neural Computing and Applications*, vol. 32, pp. 3789–3801, 2020.

[53] M. D. Incecco, S. Squartini and M. Zhong, "Transfer learning for non-intrusive load monitoring," *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 1419–1429, 2020.

[54] Q. Z. Yao, X. Zhenfan, W. Z. Wei and Y. Han, "Non-intrusive load monitoring based on deep pairwise-supervised hashing to detect unidentified appliances," *Processes*, vol. 9, no. 3, pp. 505, 2021.

[55] A. Faustine, L. Pereira and C. Klemenjak, "Adaptive weighted recurrence graphs for appliance recognition in non-intrusive load monitoring," *IEEE Transactions on Smart Grid*, vol. 12, no. 1, pp. 398–406, 2021.

[56] A. Faustine and L. Pereira, "Improved appliance classification in non-intrusive load monitoring using weighted recurrence graph and convolutional neural networks," *Energies*, vol. 13, no. 13, pp. 1–15, 2020.

[57] O. Ronneberger, P. Fischer and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," *Medical Image Computing and Computer-Assisted Intervention*, vol. 9351, pp. 234–241, 2015.