

Non-intrusive Load Monitoring Based on Regularized ResNet with Multivariate Control Chart

Cheolhwan Oh and Jongpil Jeong(✉)

Department of Smart Factory Convergence, Sungkyunkwan University, Suwon, Jangan-gu, Gyeonggi-do 16419, Republic of Korea
{dhdldzhkf13,jpjeong}@skku.edu

Abstract. With the development of industry and the spread of the Smart Home, the need for power monitoring solution technologies for effective energy management systems is increasing. Of these, non-intrusive load monitoring (NILM), is an efficient way to solve the electricity consumption monitoring problem. NILM is a technique to measure the power consumption of individual devices by analyzing the power data collected through smart meters and commercial devices. In this paper, we propose a deep neural network (DNN)-based NILM technique that enables energy disaggregation and power consumption monitoring simultaneously. Energy disaggregation is performed by learning a deep residual network for performing multilabel regression. Real-time monitoring is performed using a multivariate control chart technique using latent variables extracted through weights of the trained model. The energy disaggregation and monitoring performance of the proposed method is verified using the public NILM Electricity Consumption and Occupancy (ECO) data set.

Keywords: Non-intrusive load monitoring · Deep neural network · Multivariate statistical process control · Regularization

1 Introduction

Recently, the use of electronics in most homes and buildings has become an indispensable factor for convenience of living, and the introduction of equipment using power in factories has been increasing due to the automation of factories following industrial development. CO2 emissions increased by 43% as the total energy use from 1984 to 2004 increased by 49% due to rapid industrial growth. The analysis of the effect of the method of feedback to consumers on the use of energy was conducted by the Electric Power Research Institute (EPRI) in the United States [1]. According to the results of their research, the energy-saving effect occurs when providing energy consumption patterns to end-users. Providing detailed power consumption information for each device in real-time produced an energy-saving effect of approximately 12% on average.

© Springer Nature Switzerland AG 2020
O. Gervasi et al. (Eds.): ICCSA 2020, LNCS 12250, pp. 646–661, 2020.
https://doi.org/10.1007/978-3-030-58802-1_47

Intrusive load monitoring (ILM) monitors energy usage through smart meters connected to each device that requires monitoring. ILM measures the energy consumption directly for each device, so it has the advantage of high accuracy [2], but because each device requires the installation of a smart meter, it is difficult to manage and expensive and, thus, challenging to generalize [3]. Non-intrusive load monitoring (NILM) uses a single smart meter that captures aggregated electrical energy signals. The goal of this technique is to discover devices that contribute to aggregated signals. NILM is preferred over ILM in both academia and industry [4] because it reduces financial costs and the burden on users or homeowners to participate in the energy monitoring process.

NILM is a method of separating the total electrical load of a building measured at a single point into individual device signals using a combination of electrical collection systems and signal processing algorithms [5]. It extracts the features of the data collected by signal processing and performs load identification through the extracted features [6].

In NILM, the feature extraction process significantly influences the load identification to be performed later, so selecting and applying an appropriate feature extraction method from the collected power data is an essential task. Many studies have proposed feature extraction methods to improve the performance of load identification, contributing to the development of NILM technology [7,8].

Deep learning technology that can automatically extract features by learning data continues to develop rapidly and is being applied positively in various industries. In contrast to traditional methods of extracting direct features, deep learning models automatically extract features from raw data [9]. In the field of NILM, a deep learning approach has also been proposed, and its effectiveness has been proven [10].

In this paper, we propose an NILM framework based on deep learning algorithms. The proposed method performs energy disaggregation through multilabel regression by training a deep learning model. Moreover, by extracting the latent variable of electrical signal data as a feature through the trained model, and using this feature to draw a multivariate control chart, it provides an end-user with a monitoring dashboard system that can report real-time energy usage. Our primary contribution is extending and applying the control chart—a monitoring methodology used predominantly in process control—to the NILM field. Furthermore, we use a deep learning model already trained for energy disaggregation, reducing the required effort.

The rest of this paper is organized as follows. Section 2 discusses related work. Section 3 explains the control chart in statistical process control. Section 4 describes the proposed NILM framework. Section 5 presents experiments conducted to evaluate the performance of the proposed approach and describes the experimental results. Section 6 discusses our conclusions.

2 Related Works

2.1 Non-intrusive Load Monitoring

NILM was introduced by Hart in the 1980s [5]. Since its introduction, several NILM algorithms have been proposed to analyze data collected at low and high sampling rates. For directly estimating the power demand of each device from the aggregate power signal, an approach based on a hidden Markov model (HMM) [11] and factorial HMM (FHMM) [12] has been studied. Another powerful option for solving signal processing problems is graph signal processing (GSP) [13,14].

In previous research, both steady-state and transient state analyses were used to separate the energy consumption per device. In steady-state analysis, active power, reactive power, current waveform, and harmonic components were used as features for load disaggregation [15]. Transient energy and transient shapes were used to disaggregate the loads in transient state analysis [16,17]. Features extracted from steady-state and transient states were used in combination to increase accuracy [18,19]. The researchers relied on hand-made feature extractors such as the Fourier transform, wavelet transform, spectral-domain representation, and electrical parameters. However, these extractors are time-consuming and error-prone, creating difficulty in finding the optimal feature for disaggregation [20].

As a solution, researchers have begun using deep neural networks (DNNs) that can automatically extract features by learning the hidden patterns of raw signals. In [10], three types of DNN structures—long short-term memory (LSTM) units, a denoising autoencoder, and a regression model—were used to predict the start time, end time, and average power demand of each device. Lukas et al. [21] proposed a combination of HMM and DNN for energy disaggregation, demonstrating results that surpass FHMM. In [22], the researchers proposed an architecture with a deep recurrent LSTM network for supervised disaggregation of loads that outperformed alternatives for loads representing the periodicity of energy consumption. He et al. [23] used multiple parallel convolution layers with various filter sizes to improve the disaggregation performance of LSTM-based models. Barsim et al. [24] proposed a deep residual network-based disaggregation model and found that deeply-stacked layers could be effective for energy disaggregation. Researchers found that a convolutional neural network (CNN) architecture based on seq2seq or seq2point may be more effective in energy disaggregation [25,26]. Other researchers proposed a supervised learning method in which a 1D CNN-RNN model combining CNN and RNN is applied to NILM [27].

3 Control Chart in Statistical Process Control

Statistical process control (SPC) is a methodology that finds solutions or improvement by identifying and interpreting problems based on data, rather than monitoring the system with a guess. It is mainly used for quality control [28].

3.1 Multivariate Control Charts

A univariate control chart can be easily applied and interpreted by directly using operational data instead of a mathematical model. However, univariate SPC methods can cause erroneous results when applied to multivariate data with a high correlation between variables, and it is not efficient to draw a control chart for each variable [29].

A multivariate control chart capable of simultaneously observing changes in two or more management items is required to efficiently manage two or more related data or to simultaneously monitor a process. Several multivariate control charts have been proposed, such as the Hotelling's T^2 chart, multivariate EWMA chart, and multivariate CUSUM chart [29]. Hotelling's T^2 chart is the most widely used [30]. The T^2 statistic is calculated as follows:

$$T^2 = (x - \bar{x})^T S^{-1} (x - \bar{x}), \tag{1}$$

where $\bar{x}$ and S are the sample mean vectors and sample covariance matrices, respectively, determined from past data $X \in R^{n \times m}$ collected in the control state, in which n and m are the numbers of samples and variables. This is used as a statistic to determine the similarity between the data collected in the control state and the newly measured data. Hotelling's T^2 statistic refers to the Mahalanobis distance between historical data collected in the control state and new measurements. The upper control limit of the T^2 chart is based on the assumption of normality and follows the F distribution [28]. Equation (2) can be used to calculate the upper control limit, which is the critical value.

$$UCL_{T^2} = \frac{m(n+1)(n-1)}{n(n-m)} F_{(m,n-m,\alpha)}, \tag{2}$$

In Eq. (2), the significance level α is the type I error rate. This is the maximum allowable limit for false alarms that misjudge positive as negative. $F_{(m,n-m,\alpha)}$ is the upper α th quantile of the F-distribution with m and $(n - m)$ degrees of freedom.

However, the multivariate process control based on Hotelling's T^2 is not useful for data with many correlated variables. If there are many correlation variables, it is difficult to invert the covariance matrix S because the covariance matrix becomes nearly a singular matrix, which leads to problematic results [31]. Furthermore, including many highly correlated variables in the data may cause multicollinearity, deteriorating the ability to detect progress shifts [32]. Consequently, various latent-variable-based control charts for extracting features from raw data have been proposed.

3.2 Latent Variable Based Multivariate Control Charts

Hotelling's T^2 chart based on principal component analysis (PCA) is a representative case among latent variable-based control chart methodologies. PCA is a technique that finds axes orthogonal to each other while preserving the variance

of data as much as possible and transforms the data in the high-dimensional space into low-dimensional space without linear correlation [33]. This is similar to the T^2 chart without PCA described in Eqs. (1) and (2), except that a Q chart for performing residual analysis is added and used together. First, the data matrix X is decomposed into individual elements through PCA, and the individual elements are further divided into principal component subspaces (PCS) and residual subspaces (RS) according to the number of principal components selected, as follows:

$$X = \hat{X} + \tilde{X} = \hat{T}\hat{P}^T + \tilde{T}\tilde{P}^T = \left[\hat{T}\ \tilde{T}\right]\left[\hat{P}\ \tilde{P}\right]^T = TP^T, \tag{3}$$

where $T = \left[\hat{T}\ \tilde{T}\right]$ and $P = \left[\hat{P}\ \tilde{P}\right]$ are the score and loading matrices, respectively. $\hat{T} \in R^{n\times p}$ and $\tilde{T} \in R^{n\times m-p}$ are score matrices belonging to PCS and RS, respectively, and $\hat{P} \in R^{m\times p}$ and $\tilde{P} \in R^{m\times m-p}$ are loading matrices belonging to PCS and RS, respectively. For the number of principal components p, after drawing the scree plot, select the number of principal components corresponding to the elbow point or the number of principal components that can explain the variance as desired by the user. After determining the number of principal components p, the PCA-based T^2 statistic can be calculated, which is calculated as follows:

$$\hat{t} = x\hat{P}, \tag{4}$$

$$T^2_{PCA} = \hat{t}\hat{\Lambda}^{-1}\hat{t}^T, \tag{5}$$

Where $\hat{t}$ is the score vector of x in the PCS and $\hat{\Lambda}$ is the diagonal matrix of the largest eigenvalues of the covariance matrix of $\hat{X}$. The upper control limit of PCA-based T^2 is calculated as Eq. (6), which is almost the same as Eq. (2), as follows:

$$UCL_{T^2_{PCA}} = \frac{p(n+1)(n-1)}{n(n-p)}F_{(p,n-p,\alpha)}, \tag{6}$$

However, because the T^2_{PCA} statistic calculated through Eq. (5) uses only the information in the PCS, variations occurring in the RS may not be detected [34]. Therefore, the Q chart is additionally used to detect shifts that cannot be explained only by the information contained in PCS. The Q chart can be constructed using the residuals obtained from RS. PCA-based Q statistics monitor the squared error between the true vector x and the vector $\hat{x}$ estimated by PCA. The PCA-based Q statistic is calculated as follows:

$$Q_{PCA} = (x - \hat{x})(x - \hat{x})^T = \tilde{x}\tilde{x}^T, \tag{7}$$

Assuming the Q statistic, squared prediction error, follows a normal distribution, we can calculate the upper control limits of the Q chart with the following approximation based on the weighted chi-squared distribution [35]:

$$UCL_{Q_{PCA}} = \frac{v}{2m}\chi^2_{(2m^2/v,\alpha)}, \tag{8}$$

where m and v are the sample mean and variance of Q_{PCA}, respectively, and α is the type I error rate. This functions accurately even when the prediction error does not follow a normal distribution [36].

However, the PCA-based MSPC technique is based on the linearity of data. Therefore, using PCA as-is without removing nonlinearity does not accurately reflect the information in the data and limits the accuracy of detecting anomalies. Consequently, researchers have proposed various methodologies to reflect nonlinearity, the most being the kernel method. Researchers [37] proposed kernel PCA (KPCA), which removes nonlinearity by mapping data to a high level using a kernel function to achieve data linearity. Furthermore, it extracts latent variables considering nonlinearity by applying PCA to linearized data. KPCA is favored because it is simple to use and can adequately consider the nonlinearity of data. In the field of process monitoring, several studies extracted latent variables using KPCA and applied to multivariate control charts, and their effectiveness has been proven [38–40].

PCA- and KPCA-based multivariate control charts are used to monitor the system that generates X data by identifying latent variables representative of the relationship of X and obtaining a control limit. However, in several cases, we prefer to monitor not X but the output Y produced by the system, for which partial least squares (PLS) is the most commonly used alternative to PCA. PLS shares similarities with PCA. Whereas PCA extracts latent variables that maximize the variance of the linear combination of X, PLS extracts latent variables that maximize the covariance between the linear combination of X and Y [41].

Just as PCA uses a matrix X consisting of measurements of process variables to monitor process variables, PLS-based control charts can also monitor quality variables using matrix Y consisting of measurements of quality variables. However, PLS, like PCA, cannot consider the nonlinearity of data. Therefore, KPLS-based methodologies for considering nonlinearity have been proposed. Furthermore, PLS- and KPLS-based multivariate control chart techniques have been proposed to monitor the status of the products produced during the process [42].

Recently, a method of applying the latest deep learning method to extract latent variables for use in a multivariate control chart has been proposed. The authors [43] proposed a method of extracting latent variables and applying them to a multivariate control chart using an unsupervised learning algorithm, one of the deep learning methods. Latent variables were extracted using a variational autoencoder model capable of extracting the feature of the independent variable X, confirming that the performance of the proposed method is superior to that of the existing PCA- and KPCA-based multivariate control charts.

4 Proposed Method

This section explains the proposed method. We propose an NILM methodology based on DNN algorithms. The primary contribution is an energy disaggregation and real-time monitoring dashboard based on the training of a single neural network model.

The proposed method is divided into three keywords. First, sequence-to-point learning corresponds to the form of the input/output data of a neural network model for performing energy disaggregation. Second, the regularized residual network corresponds to the architecture of a DNN performing NILM. Regularization is used to extract latent variables to be used in multivariate control charts. The last keyword, latent-variable-based multivariate control charts, aims to provide real-time monitoring dashboards to end-users. The latent variables needed to implement a multivariate control chart are extracted from the residual network for energy disaggregation and consumption prediction—no additional process is used to extract latent variables for implementing a multivariate control chart.

4.1 Sequence-to-Point Learning

We generalized the energy disaggregation problem of NILM as a prediction problem using a neural network model into a regression model that predicts the energy consumption of individual electronic devices $Y \in R^{n \times k}$ using data $X \in R^{n \times m}$ collected from the main power line. In this model, n is the number of data records collected according to the sampling rate of the smart meter and m is the number of features of the electrical data set collected and stored by the smart meter installed on the main power line. These features typically include voltage and current, phase, power factor, and I-V trajectory [6]. k is the number of electronic devices requiring prediction of energy consumption through a regression model.

For defining the regression problem for energy disaggregation as a sequence-to-point neural network model, the original electrical data matrix X is sliced into a specific sequence length l, which is expressed as a three-dimensional array W through the following procedure:

$$X = \{x_1, x_2, \cdots, x_n\}, \tag{9}$$

$$S_{i:j} = \{x_i, x_{i+1}, \cdots, x_j\}, 1 \leq i \leq j \leq n, \tag{10}$$

$$W = \{S_{1:l}, S_{2:l+1}, \cdots, S_{n-l+1:n}\}, \tag{11}$$

A method of improving the energy disaggregation performance by predicting the energy consumption Y corresponding to the midpoint of an individual sequence was proposed [25], but because this method predicts the midpoint of the sequence, a time delay of $\frac{l}{2}$ occurs. The lower the sampling rate and longer the sequence length l, the more challenging the model becomes for real-time monitoring. Therefore, we designed the input/output structure of the model to predict energy consumption corresponding to the endpoint of each sequence rather than the midpoint.

4.2 Regularized ResNet Based Multivariate Control Charts

The primary feature of the residual network is the residual shortcut connection between consecutive convolutional layers. The difference between a residual network and a typical convolutional network, such as a fully convolutional network (FCN), is that a linear shortcut residual connection is added to connect the input and output of the residual block [44]. Through this connection, the residual network robustly adapts to the degradation problem, which was one of the chronic problems in DNNs, facilitating the neural network training for feature extraction by building the neural network into a deeper structure.

Therefore, we used a deep residual network to sufficiently extract features from the electrical data collected by the smart meter. The structure of the model in this study consists of six residual blocks, each composed of three layers; the end of the model is composed of a global average pooling (GAP) layer and regularized linear activation function to perform multilabel regression.

In general neural network architectures, it is common not to apply regularization to the layer that performs classification and regression using latent variables extracted through the previous layer. However, because we need to monitor the energy consumption Y of individual electronic devices through latent variables extracted from the GAP layer, the GAP layer must contain enough information about Y. Accordingly, we apply strong regularization to the linear activation function layer that performs regression such that the latent variable extracted from the GAP layer does not depend on the weight of the linear activation function layer. In this study, regularization was performed through L2-norm and Max-norm [45].

The overall structure of the model is illustrated in Fig. 1. Researchers demonstrated that a residual network designed with this architecture and parameters is effective in extracting features from time-series data [46] and provided strong guidelines for designing a residual network for feature extraction of time-series data. Algorithm 1 is the pseudocode for a method of monitoring energy consumption using a multivariate control chart.

5 Experiment and Performance Analysis

5.1 Experimental Data

We used the Electricity Consumption and Occupancy (ECO) data set for our experiment, which was collected from six swiss households over eight months [47] at a sampling rate of 1 Hz.

For the main meter, the ECO dataset has 16 metered variables, including total power consumption, three-phase power, neutral current, three-phase current, three-phase voltage, and phase angles of $V12$, $V13$, $I1{-}V1$, $I2{-}V2$, and $I3{-}V3$. The sub-metered data consists of several types of appliances. Because the data collected from the main meter in the ECO dataset is multivariate with 16 variables, it is suitable for use in this study, which proposes a methodology for monitoring energy usage based on a multivariate control chart.

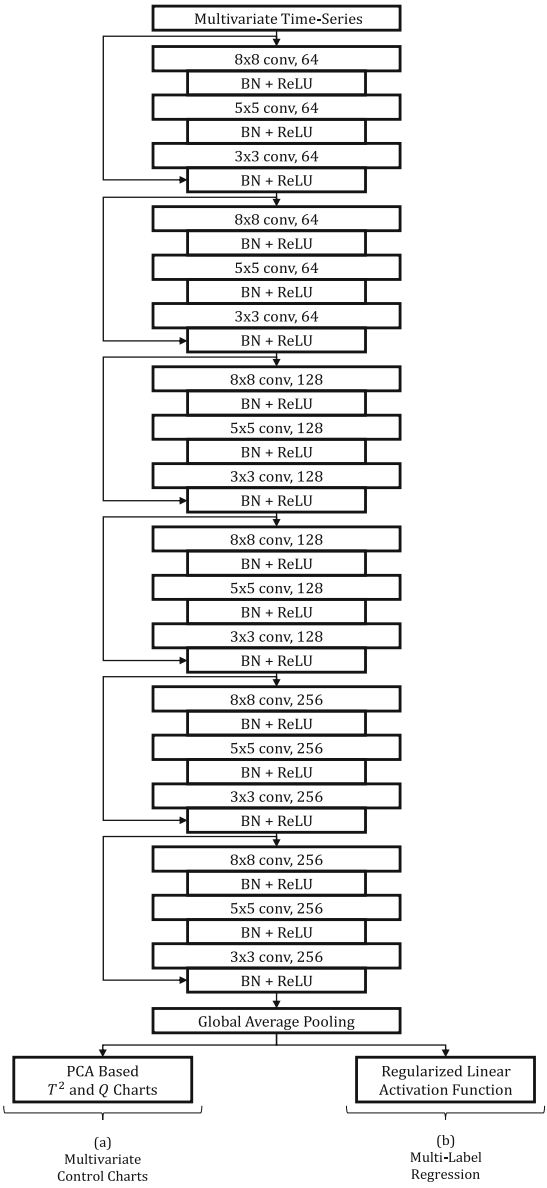


Fig. 1. The architecture of the residual network for energy disaggregation. In (a), latent variables are extracted., and in (b), multi-label regression is performed to predict the energy consumption of each electronic device

Algorithm 1: Multivariate Control Charts based on Latent Variable extracted from Regularized ResNet

Input: training set $X_{tr} \in R^{n \times m}$ and $Y_{tr} \in R^{n \times k}$
Input: test set $X_{te} \in R^{r \times m}$ and $Y_{te} \in R^{r \times k}$
Input: sequence length l
Input: Explained variance v
Input: Regularization parameter λ and c
Input: Type 1 error rate α
$W_{tr} \leftarrow$ Reshaping data X_{tr} by l through equations (11)
`/* Train predictive models */`
Let $Residual\,Block_i$, where $i = \{1, 2, \cdots, d\}$
$\theta \leftarrow$ Initialize the parameters of the predictive models
while *Update until* θ *converges* **do**
  $Z_{tr_1} = Residual\,Block_1(W_{tr})$
  $\cdots$
  $Z_{tr_d} = Residual\,Block_d(Z_{tr_{d-1}})$
  $Z_{tr_{gap}} = GAP(Z_{tr_d})$ `// 3d array to 2d matrix`
  $H = Linear\,Activation\,Function(Z_{tr_{gap}},\ L2\ = \lambda,\ Max - norm = c)$
  $Cost = Mean\,Squared\,Error(H, Y_{tr})$
  $\theta \leftarrow$ Update the parameters through the gradient of $Cost$
end
`/* Set threshold for control charts based on` α `*/`
Train PCA model through $Z_{tr_{gap}}$, The number of PC is selected by v.
$UCL_{T^2_{PCA}} \leftarrow$ Calculate the control limit of the T^2 chart through Eq(6)
$UCL_{Q_{PCA}} \leftarrow$ Calculate the control limit of the Q chart through Eq(8)
`/* Monitoring for energy consumption alarm */`
for *j = 1 to r-l+1* **do**
  $S_{j:l+j-1} \leftarrow$ Reshaping data X_{te} through equations (10)
  $z_{te_{gap}} = GAP(S_{j:l+j-1})$ `// 2d matrix to 1d vector`
  $t_{te} = PCA(z_{te_{gap}})$ `// Calculate score vector`
  $T^2_{PCA} \leftarrow$ Calculate T^2 statistics based on t_{te} through Eq(5)
  $Q_{PCA} \leftarrow$ Calculate Q statistics based on t_{te} through Eq(7)
  if $T^2_{PCA} \geq UCL_{T^2_{PCA}}$ *and* $Q_{PCA} \geq UCL_{Q_{PCA}}$ **then**
    energy consumption alarm
  end
end

In the ECO data set, the number and type of electronic devices that collect data for each household differ, as are the ratios of the total amount of power consumed by each electronic device to the total energy consumption of each household. Of the six houses, house2 collected data from the largest number of household appliances, and electricity consumption covered by the smart plugs was also highest. Therefore, we conducted an experiment using data collected from house 2. House2 collects data from 12 individual electronic devices.

In this paper, we experiment using the data collected in January 2013 from the total data from House 2. The stove corresponding to the data from the

tenth electronic device is not continuously collected. Therefore, the experiment is performed only on 11 data points, excluding the stove. The periods used are January 1–21, 2013, for training data, and January 22–28, 2013, for test data. The model was trained using data for three weeks, and the trained model was tested using data collected during the following one week.

5.2 Parameter Configuration

Each residual block is first composed of three convolutions whose output is added to the residual block's input and then fed to the next layer. The number of filters for convolutions is fixed at 64, 128, and 256, with a ReLU activation function preceded by a batch normalization operation. In each residual block, the filter's length is set to 8, 5, and 3, respectively, for the first, second, and third convolution. The stride is fixed at 1. The padding is set so that the input and output of the layer are equivalent. For residual network optimization, we used Adam optimizer with a learning rate of 0.001 as the optimization Algorithm, and the loss was set to the mean square error (MSE). The training was performed for 1000 epochs, and the batch size was set to 4096. The validation data set used as much as 10% of the training data set. λ and c are set to 0.1 and 1, respectively.

The number of principal components of PCA performed to implement T^2 and Q charts using latent variables extracted from the last GAP layer of the residual network was selected based on explained variance, and the threshold was set to 80%. Furthermore, the type I error rate of the control limit α was set to 5%.

The sequence length l for performing sequence-to-point learning was set to 60 in the experiment. We use the data collected from x_{t-59} to x_t to predict the energy consumption y_t.

5.3 Performance Evaluation and Analysis

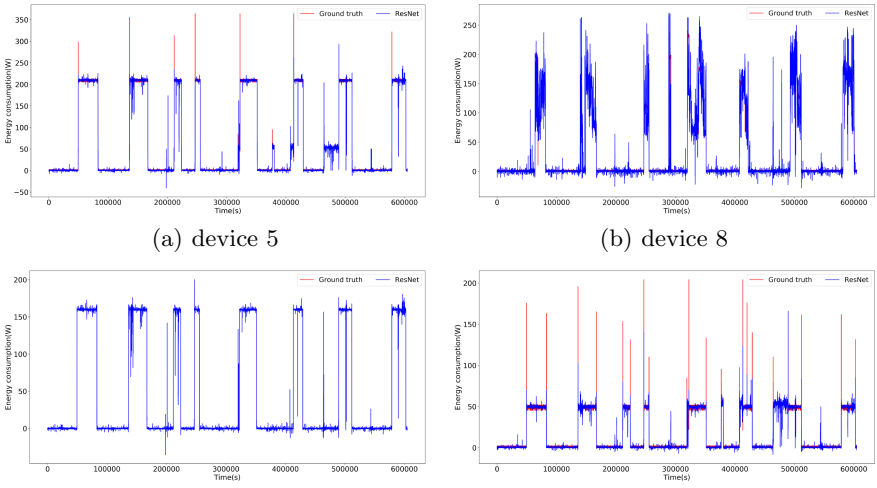
We use four metrics to evaluate the performance of multi-label regression performed by the residual network from various perspectives. The metrics we used are Root Mean Square Error (RMSE), Signal Aggregate Error (SAE), Normalized Disaggregation Error (NDE), and Accuracy.

First, we evaluate the performance of the residual network using RMSE metrics, which are commonly used to evaluate the predictive performance of regression models. RMSE $= \sqrt{\frac{1}{T}\sum_{t=1}^{T}(\hat{y_t}^i - y_t^i)^2}$. Where, $\hat{y_t}^i$ and y_t^i are prediction and ground truth of the i-th electronic device at the t-th time step, respectively.

The second metric, SAE, is known to evaluate the model in terms of the total error in energy over a period of time. SAE $= \frac{\left|\sum_{t=1}^{T}\hat{y_t}^i - \sum_{t=1}^{T} y_t^i\right|}{\sum_{t=1}^{T} y_t^i}$. In contrast to the SAE, which focuses on energy consumption over a period of time, the third metric, NDE, is a measure of whether the predictive model predicts energy consumption well at every time point. NDE $= \frac{\sum_{t=1}^{T}(\hat{y_t}^i - y_t^i)^2}{\sum_{t=1}^{T}(y_t^i)^2}$. The fourth metric, accuracy, evaluates whether the total energy is properly classified regardless of

whether each electronic device is a low-power or high-power device [48]. Accuracy $= 1 - \frac{\sum_{t=1}^{T}\sum_{i=1}^{k}\left|\hat{y_t}^i - y_t^i\right|}{2\sum_{t=1}^{T}\sum_{i=1}^{k} y_t^i}$. We compare our experimental results with those of previously proposed researchers to evaluate the performance of our proposed model.

We conducted a comparative evaluation using the methods suggested by the researchers in [14] and their experimental results. They proposed a method to improve energy disaggregation performance by extracting patterns and features from multivariate data collected from the main meter using a graphical modeling approach, a spatiotemporal pattern network (STPN). Furthermore, they provide the experimental results of FHMM, combinatorial optimization (CO), and probabilistic finite-state automaton (PFSA), which are general methodologies for performing energy disaggregation in NILM, along with the experimental results of their proposed method. They conducted experiments using training and test data sets collected during the same period used in our study. The performance metrics RMSE, AE, NDE, and accuracy for the five energy disaggregation models are listed in Table 1. Based on the experiment, the prediction of the residual network using sequence-to-point learning outperforms other methods (Fig. 2).



(a) device 5 (b) device 8

(c) device 11 (d) device 12

Fig. 2. Example of the difference between the ground truth and prediction for each electronic device predicted by our proposed residual network.

For evaluating the monitoring performance of a multivariate control chart, we confirm whether energy consumption rapidly increased when statistics exceeded the control limit. By identifying the level at which the multivariate control chart caused a false alarm, we evaluate the performance of the control chart drawn with the latent variable extracted by our proposed method. When the statistics exceed the control limit, if the statistics outside the control limit are higher

Table 1. The device-level RMSE, SAE, NDE and Acc. Best results are shown in bold.

Method	Metrics	Device number											Sum	Acc.
		1	2	3	4	5	6	7	8	9	11	12		
FHMM	RMSE	1.6	203.0	30.8	45.4	87.1	37.4	170.0	59.6	19.2	68.7	27.1	749.9	0.56
	SAE	0.3	0.1	56.1	0.1	0.5	0.1	3.9	0.3	0.3	0.5	0.3	62.5	
	NDE	1.0	0.9	76.1	0.9	0.6	0.7	4.1	0.8	1.2	0.6	0.8	87.7	
CO	RMSE	2.2	197.4	42.1	72.2	94.2	71.4	138.0	60.3	30.3	64.5	29.4	802.0	0.44
	SAE	0.5	0.1	92.4	0.1	0.6	0.3	2.6	0.1	1.5	0.4	0.3	98.9	
	NDE	1.9	0.9	142.5	2.3	0.7	2.7	2.7	0.8	3.0	0.6	1.0	159.1	
PFSA	RMSE	1.1	139.3	4.3	41.5	36.1	35.9	64.9	35.3	15.0	30.4	12.4	416.2	0.70
	SAE	0.026	0.070	1.314	0.018	0.034	**0.006**	0.159	0.046	0.142	0.028	0.052	1.895	
	NDE	0.504	0.443	**1.497**	0.772	0.100	0.676	0.601	0.265	0.732	0.124	0.171	5.885	
STPN	RMSE	1.1	68.6	6.3	38.7	8.4	35.4	57.4	24.2	15.5	5.8	6.1	267.5	0.84
	SAE	0.022	**0.022**	1.708	**0.003**	0.016	0.032	0.596	**0.001**	**0.087**	0.006	0.056	2.549	
	NDE	0.522	0.108	3.218	0.673	0.005	0.655	0.470	0.124	0.781	0.004	0.041	6.601	
ResNet	RMSE	**1.0**	**29.8**	**5.2**	**29.3**	**6.2**	**25.1**	**25.6**	**23.7**	**12.4**	**5.3**	**3.2**	**166.8**	**0.93**
	SAE	**0.005**	0.049	**0.839**	0.008	**0.006**	0.040	**0.048**	0.003	0.159	**0.003**	**0.007**	**1.167**	
	NDE	**0.489**	**0.020**	2.181	**0.386**	**0.003**	**0.330**	**0.093**	**0.119**	**0.501**	**0.003**	**0.011**	**4.136**	

based on the T-test than the energy consumption of the previous l, they are classified as outliers. The monitoring performance is evaluated by calculating the false positive rate (FPR) (i.e., one-specificity, false alarm rate) [49]. FPR decreased from 13.8% for the general PCA-based T^2 and Q chart to 9.5% for ResNet without regulation and 4.9% for ResNet with regularization.

6 Conclusion

In this paper, we proposed a framework to simultaneously perform energy disaggregation and multivariate control chart-based monitoring as a method of extracting features through a sequence-to-point learning-based residual network from multivariate data collected from the main meter. Our primary contribution is achieving both tasks simultaneously by training only one model. Moreover, we proposed a framework for applying multivariate control chart techniques, commonly used in process management, to NILM. Experimental results demonstrate that the neural network model trained by our proposed method can extract latent variables sufficient for the monitoring of energy consumption of electronic devices and energy disaggregation. In this study, the performance was evaluated only with FPR because the energy increase event could not be separately extracted from the energy data. Furthermore, because experiments have not been applied through all regulations, it was not possible to identify which regulatory methods and parameters are optimal. Therefore, future research should apply additional regulatory methodologies to data that can accurately specify abnormal events.

Acknowledgments. This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support

program (IITP-2020-2018-0-01417) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation).

References

1. Neenan, B., Robinson, J., Boisvert, R.N.: Residential electricity use feedback: a research synthesis and economic framework. Electric Power Research Institute, March 2009
2. Strbac, G.: Demand side management: benefits and challenges. Energy Policy **36**(12), 4419–4426 (2008)
3. Suzuki, K., Inagaki, S., Suzuki, T., Nakamura, H., Ito, K.: Nonintrusive appliance load monitoring based on integer programming. In: 9th 2008 SICE Annual Conference, pp. 2742–2747. IEEE (2008)
4. Hosseini, S.S., Agbossou, K., Kelouwani, S., Cardenas, A.: Non-intrusive load monitoring through home energy management systems: a comprehensive review. Renew. Sustain. Energy Rev. **79**, 1266–1274 (2017)
5. Hart, G.W.: Nonintrusive appliance load monitoring. Proc. IEEE **80**(12), 1870–1891 (1992)
6. Ruano, A., Hernandez, A., Ureña, J., Ruano, M., Garcia, J.F.: NILM techniques for intelligent home energy management and ambient assisted living: a review. Energies **12**(11), 2203 (2019)
7. Sadeghianpourhamami, N., Ruyssinck, J., Deschrijver, D., Dhaene, T., Develder, C.: Comprehensive feature selection for appliance classification in NILM. Energy Build. **151**, 98–106 (2017)
8. Wang, A.L., Chen, B.X., Wang, C.G., Hua, D.: Non-intrusive load monitoring algorithm based on features of V-I trajectory. Electr. Power Syst. Res. **157**, 134–144 (2018)
9. Bengio, Y., Courville, A., Vincent, P.: Representation learning: a review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell. **35**(8), 1798–1828 (2013)
10. Kelly, J., Knottenbelt, W.: Neural NILM: deep neural networks applied to energy disaggregation. In: Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments, pp. 55–64. ACM (2015)
11. Makonin, S., Popowich, F., Bajić, I.V., Gill, B., Bartram, L.: Exploiting HMM Sparsity to perform online real-time nonintrusive load monitoring. IEEE Trans. Smart Grid **7**(6), 2575–2585 (2015)
12. Lange, H., Bergés, M.: Efficient inference in dual-emission FHMM for energy disaggregation. In: Workshops at the Thirtieth AAAI Conference on Artificial Intelligence (2016)
13. He, K., Stankovic, L., Liao, J., Stankovic, V.: Non-intrusive load disaggregation using graph signal processing. IEEE Trans. Smart Grid **9**(3), 1739–1747 (2018)
14. Liu, C., Akintayo, A., Jiang, Z., Henze, G.P., Sarkar, S.: Multivariate exploration of non-intrusive load monitoring via spatiotemporal pattern network. Appl. Energy **211**, 1106–1122 (2018)
15. Nakano, Y., Murata, H.: Non-intrusive electric appliances load monitoring system using harmonic pattern recognition-trial application to commercial building. In: International Conference on Electrical Engineering, Hong Kong, China (2007)

16. Chen, K.-L., Chang, H.-H., Chen, N.: A new transient feature extraction method of power signatures for nonintrusive load monitoring systems. In: 2013 IEEE International Workshop on Applied Measurements for Power Systems (AMPS), pp. 79–84. IEEE (2013)
17. Chang, H.-H., Lian, K.-L., Su, Y.-C., Lee, W.-J.: Power-spectrum-based wavelet transform for nonintrusive demand monitoring and load identification. IEEE Trans. Ind. Appl. **3**(50), 2081–2089 (2014)
18. Liang, J., Ng, S.K.K., Kendall, G., Cheng, J.W.M.: Load signature study-part I: basic concept, structure, and methodology. IEEE Trans. Power Deliv. **25**(2), 551–560 (2009)
19. Liang, J., Ng, S.K.K., Kendall, G., Cheng, J.W.M.: Load signature study-part II: disaggregation framework, simulation, and applications. IEEE Trans. Power Deliv. **25**(2), 561–569 (2010)
20. Sirojan, T., Phung, B.T., Ambikairajah, E.: Deep neural network based energy disaggregation. In: 2018 IEEE International Conference on Smart Energy Grid Engineering (SEGE), pp. 73–77. IEEE (2018)
21. Mauch, L., Yang, B.: 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), pp. 2384–2388. IEEE (2016)
22. Mauch, L., Yang, B.: 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), pp. 63–67. IEEE (2015)
23. He, W., Chai, Y.: An empirical study on energy disaggregation via deep learning. In: 2nd International Conference on Artificial Intelligence and Industrial Engineering (AIIE 2016). Atlantis Press (2016)
24. Barsim, K.S., Yang, B.: On the feasibility of generic deep disaggregation for single-load extraction. arXiv preprint arXiv:1802.02139 (2018)
25. Zhang, C., Zhong, M., Wang, Z., Goddard, N., Sutton, C.: Sequence-to-point learning with neural networks for non-intrusive load monitoring. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
26. Chen, K., Wang, Q., He, Z., Chen, K., Hu, J., He, J.: Convolutional sequence to sequence non-intrusive load monitoring. J. Eng. **2018**(17), 1860–1864 (2018)
27. Çavdar, İ.H., Faryad, V.: New design of a supervised energy disaggregation model based on the deep neural network for a smart grid. Energies **12**(7), 1217 (2019)
28. Montgomery, D.C.: Introduction to Statistical Quality Control. Wiley, New York (2007)
29. Lowry, C.A., Montgomery, D.C.: A review of multivariate control charts. IIE Trans. **27**(6), 800–810 (1995)
30. Hotelling, H.: Multivariate Quality Control. Techniques of Statistical Analysis. McGraw-Hill, New York (1947)
31. Seborg, D.E., Mellichamp, D.A., Edgar, T.F., Doyle III, F.J.: Process Dynamics and Control. Wiley, New York (2010)
32. Ku, W., Storer, R.H., Georgakis, C.: Disturbance detection and isolation by dynamic principal component analysis. Chemom. Intell. Lab. Syst. **30**(1), 179–196 (1995)
33. Wold, S., Esbensen, K., Geladi, P.: Principal component analysis. Chemom. Intell. Lab. Syst. **2**(1–3), 37–52 (1987)
34. Mastrangelo, C.M., Runger, G.C., Montgomery, D.C.: Statistical process monitoring with principal components. Qual. Reliab. Eng. Int. **12**(3), 203–210 (1996)

35. Box, G.E.P., et al.: Some theorems on quadratic forms applied in the study of analysis of variance problems, I. Effect of inequality of variance in the one-way classification. Ann. Math. Stat. **25**(2), 290–302 (1954)
36. van Sprang, E.N.M., Ramaker, H.-J., Westerhuis, J.A., Gurden, S.P., Smilde, A.K.: Critical evaluation of approaches for on-line batch process monitoring. Chem. Eng. Sci. **57**(18), 3979–3991 (2002)
37. Schölkopf, B., Smola, A., Müller, K.-R.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput. **10**(5), 1299–1319 (1998)
38. Lee, J.-M., Yoo, C., Choi, S.W., Vanrolleghem, P.A., Lee, I.-B.: Nonlinear process monitoring using kernel principal component analysis. Chem. Eng. Sci. **59**(1), 223–234 (2004)
39. Ge, Z., Yang, C., Song, Z.: Improved kernel PCA-based monitoring approach for nonlinear processes. Chem. Eng. Sci. **64**(9), 2245–2255 (2009)
40. Mansouri, M., Nounou, M., Nounou, H., Karim, N.: Kernel PCA-based GLRT for nonlinear fault detection of chemical processes. J. Loss Prev. Process Ind. **40**, 334–347 (2016)
41. Geladi, P., Kowalski, B.R.: Partial least-squares regression: a tutorial. Anal. Chim. Acta **185**, 1–17 (1986)
42. Yi, J., Huang, D., He, H., Zhou, W., Han, Q., Li, T.: A novel framework for fault diagnosis using kernel partial least squares based on an optimal preference matrix. IEEE Trans. Ind. Electron. **64**(5), 4315–4324 (2017)
43. Lee, S., Kwak, M., Tsui, K., Kim, S.B.: Process monitoring using variational autoencoder for high-dimensional nonlinear processes. Eng. Appl. Artif. Intell. **83**, 13–27 (2019)
44. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
45. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)
46. Wang, Z., Yan, W., Oates, T.: Time series classification from scratch with deep neural networks: a strong baseline. In: International Joint Conference on Neural Networks (IJCNN), pp. 1578–1585. IEEE (2017)
47. Beckel, C., Kleiminger, W., Cicchetti, R., Staake, T., Santini, S.: The ECO data set and the performance of non-intrusive load monitoring algorithms. In: Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings, pp. 80–89 (2014)
48. Kolter, J.Z., Johnson, M.J.: REDD: a public data set for energy disaggregation research. In: Proceedings of the SustKDD Workshop on Data Mining Applications in Sustainability (2011)
49. Zheng, D., Li, F., Zhao, T.: Self-adaptive statistical process control for anomaly detection in time series. Expert Syst. Appl. **57**, 324–336 (2016)