

Energy demand pattern analysis in South Korea using hidden Markov model-based classification

Jaeyong Lee | Beom Seuk Hwang 

Department of Applied Statistics, Chung-Ang University, Seoul, Korea

Correspondence

Beom Seuk Hwang, Department of Applied Statistics, Chung-Ang University, 84 Heukseok-ro, Dongjak-gu, Seoul, 06974, Korea.

Email: bshwang@cau.ac.kr

Funding information

National Research Foundation of Korea, Grant/Award Number: NRF-2019R1C1C1011710; Korea Institute of Energy Technology Evaluation and Planning (KETEP) and the Ministry of Trade, Industry & Energy (MOTIE) of the Republic of Korea, Grant/Award Number: 20199710100060

Abstract

Understanding energy demand patterns in the residential sector is crucial for improving energy efficiency through demand-side management. Load curve classification is a useful method for analyzing energy demand patterns. In this paper, we employ a hidden Markov model (HMM)-based classification to residential load curves in South Korea. We also investigate how the number of hidden states affects classification performance by allowing HMM to train with a different number of hidden states for each class. We compare our HMM-based method with several state-of-the-art models and find that it outperforms other competing models in multiple datasets. Additionally, we use the fitted HMM model to make inferences about the load curves, gaining deeper insights into energy demand patterns.

KEYWORDS

energy demand pattern, hidden Markov model, residential load curve, time series classification

JEL CLASSIFICATION

C15, C22, Q41

This is an open access article under the terms of the [Creative Commons Attribution-NonCommercial-NoDerivs](https://creativecommons.org/licenses/by-nc-nd/4.0/) License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

© 2024 The Author(s). *Asian Economic Journal* published by East Asian Economic Association and John Wiley & Sons Australia, Ltd.

Received: 13 May 2023; Accepted: 29 June 2024

1 | INTRODUCTION

Understanding demand patterns is fundamental in numerous economic contexts, from resource allocation to policy formulation. Energy demand, particularly in the residential sector, is a significant component of this framework, accounting for up to half of the total energy demand in many countries. Analyzing these patterns yields crucial insights not only for energy producers and consumers but also for economic planners and policymakers aiming to enhance efficiency and sustainability. Given the recent advancements in demand-side management (DSM) strategies, the characterization of demand patterns has gained heightened importance. The primary objective of DSM is to provide cost-effective and reliable energy to customers by optimizing electricity usage at the user end, yielding benefits for both consumers and producers (Premkumar et al., 2022). Moreover, energy demand pattern analysis influences housing and urban development by informing infrastructure planning and supporting smart city initiatives, promoting energy-efficient housing to reduce overall consumption and costs. In energy pricing and subsidies, it aids in designing fair tariff structures and tailoring subsidies for energy-efficient appliances. Environmental policies benefit from emission reduction efforts and energy conservation programs, promoting renewable energy and efficient technologies in homes.

Since understanding demand patterns is essential in implementing effective DSM, there have been various attempts at characterizing load curves. Balasubramanian and Balachandra (2021) used a simulation-based clustering of load curves and demonstrated its effectiveness. Within the residential sector, home appliances contribute significantly to energy consumption. According to Firth et al. (2008), the quantity and types of these appliances within a household exert a notable influence on electricity demand. Several studies have focused on investigating the effect of seasonal appliances, such as air conditioners, on household electricity consumption. Hu et al. (2019) suggested that the use of air conditioners or other weather-related devices is on an increasing trend. As such, seasonal home appliances can be useful in understanding energy demand patterns of the residential sector.

In this study, we classify the electricity load curves of South Korean households based on the types and quantities of home appliances. Four types of home appliances are considered: seasonal, video/audio, kitchen, and general devices. A survey was conducted to gather information on the specific devices owned for each of the four types. We aggregated the number of devices for each type and used them to analyze the electricity load curves of each household. Given that energy demand is likely to be lower for households with fewer electronic devices and higher for those with more, we divided the distribution of the number of possessions into two, creating two distinct classes for our analysis. This classification provides insights into household energy behavior, which is vital for developing targeted DSM strategies and economic policies.

Time series classification is a task associated with classifying each time series into a known label and there have been various methods developed over the years. The k -nearest neighbors (KNN) algorithm is one of the most widely used distance-based models and is considered the baseline model for time series classification. The KNN classifies data into a class similar to the one from training data, where it determines the class of the KNNs based on a measure of distance. A popular metric to determine the nearest neighbors in time series classification is dynamic time warping (DTW), which is the smallest distance of alignment between time series. In addition to DTW, numerous alternative elastic distance metrics (distance measures that compensate for potential misalignment between series) have been proposed: for example, Elastic Ensemble (Tan et al., 2020). The distance-based algorithm has the advantage that it does not require assumptions about the data, unlike parametric models. However, one of the disadvantages of the KNN is that it is highly sensitive to irrelevant features and the scale of the data, and the calculations tend to take a long time for large k .

There are several feature-based models that have shown promising results in time series classification. The bag-of-SFA-symbols in vector space (BOSSVS) proposed by Schäfer (2015) is a time-efficient time series classification model that is significantly faster than the k -nearest-neighbor classifier. The algorithm transforms the time series into a histogram of symbols by the symbolic Fourier approximation (SFA). Subsequently, the TF-IDF (term frequency-inverse document frequency) model is employed to compute the cosine similarity within the vector space model. The classification decision for a given time series is made based on the class that exhibits the highest cosine similarity. Baydogan et al. (2013) proposed a time series classification algorithm based on the bag-of-features approach (TSBF), which characterizes complex objects by generating feature vectors of subobjects. The algorithm initially extracts subsequences of random lengths from random locations of the time series. Features such as the mean and standard deviation of each subsequence are obtained. Each subsequence is then partitioned into several intervals, and features are computed from each interval. The algorithm employs a random forest classifier using these features to classify the time series. Recently, alternative pipelines of feature extractor and classifier have been developed, for example, time series feature extraction based on scalable hypothesis tests (Christ et al., 2018).

Deep learning models have recently become one of the most successful fields of study, and are increasingly used for time series classification (Ismail Fawaz et al., 2019). Wang et al. (2017) proposed the fully convolutional network (FCN) which can be a simple but strong baseline for time series classification. The FCN has the advantage that it does not necessitate complicated preprocessing on the raw time series. The FCN inspired and produced many other variants for time series classification. Karim et al. (2017) extended the FCN by augmenting it with a long short-term recurrent neural network submodule, called LSTM-FCN. This modification was made to enable the

LSTM-FCN to capture temporal dependencies in sequential data. Elsayed et al. (2019) proposed another variant of the FCN that replaces the LSTM submodule with a gated recurrent unit (GRU), called GRU-FCN. The architecture of GRU is even simpler than LSTM and it can outperform LSTM-FCN in many time series classification datasets.

Ismail Fawaz et al. (2019) claimed that time series classification task is similar to computer vision task where the only difference is the dimension of the data. They also proposed InceptionTime (IT), which is an ensemble of inception networks each with different initialized weights. The main idea of inception network is to apply various filters to the time series simultaneously which allows the model to extract relevant features from the series. According to Tang et al. (2020) the receptive field (RF) size is one of the most important factors to consider for one-dimensional convolutional neural network (1D-CNN) used in univariate time series classification. Since the RF size greatly affects the performance, several studies have been made on how to choose the right size. Tang et al. (2020) proposed a model with an omni-scale block for 1D-CNN, called omni-scale convolutional neural network (OS-CNN) that can automatically set the RF size. Due to its ability to find an appropriate RF size effectively, a simple 1D-CNN with an OS-block can achieve great performance across multiple benchmark datasets. More recently, Dempster et al. (2020) introduced ROCKET (random convolutional kernel transform) as a method for transforming time series using a considerable number of random convolutional kernels, achieving state-of-the-art accuracy in time series classification. Subsequently, Dempster et al. (2020) reformulated ROCKET into a new approach named MINIROCKET, which can be up to 75 times faster than ROCKET on larger datasets while maintaining essentially the same level of accuracy. Also, the time series transformer (TST), introduced by Zerveas et al. (2021), is a transformer-based framework designed for unsupervised representation learning of multivariate time series. This framework surpasses the performance of state-of-the-art supervised methods in tasks related to time series regression and classification. Despite the evident success of deep learning models in various fields, they have a few disadvantages. One of the most notable disadvantages of deep learning models is that they require a large amount of data to train properly.

In this study, we consider a probabilistic model called the hidden Markov model (HMM), which models a system in which a series of observations are derived from a series of unobservable hidden states, assumed to be a Markov process. Several advantages of HMMs exist over the methods discussed above. The HMM can learn the sequence of hidden states through an observation sequence and make probabilistic statements about each state through a transition probability matrix and emission probability matrix. This approach allows a more thorough understanding of the observation sequence and allows a probabilistic approach to the problem. Moreover, they have strong statistical foundations and are robust to new data. They are easy to train and test on new data

once the HMM has been initialized (Zucchini et al., 2017). Finally, they have well-established training algorithms, such as the Baum–Welch algorithm, which guarantees convergence to a local optimum. Another advantage of the HMM is that it is not sensitive to the sample size. When examining relatively small data, various studies, including those by Andersson (2017), and Dofadar et al. (2021), have consistently reported that HMM approaches exhibit exceptional performance. They affirmed that achieving optimal performance does not necessarily require a large sample size; in certain instances, having as few as 40 or 80 subjects suffices. The disadvantage of the HMM is that it usually has numerous parameters to learn, which is computationally expensive and time-consuming (Zucchini et al., 2017).

The HMM is a popular model for time series data and has broad applicability in various fields. Dofadar et al. (2021) employed the HMM to forecast confirmed cases and deaths from the coronavirus disease 2019 in Bangladesh. The HMM is also remarkably useful in classification problems. Andersson (2017) classified whether the DSM has been utilized based on electricity load curves using HMM-based classification.

The fundamental premise of HMMs is that observation times are discrete and regular. However, this assumption is frequently impractical in clinical trials and observational studies where subjects are intermittently observed, and the observation times are irregular or continuous. In addressing the time-offset issue, Axelrod and Maison (2004) integrated HMMs with DTW procedures. The combination of DTW with multibranch HMM models proved successful in reducing the error rate in large vocabulary speech recognition. Recently, a diverse range of HMMs has emerged to address the time-offset challenge. Guédon (2003) proposed a hidden semi-Markov model, incorporating a positive integer random variable to define state dwell-time and separately modeling the transition probability matrix of the embedded Markov chain, to allow irregular observation times. Various formulations of continuous-time HMMs have also been presented. These models, known for their high flexibility, are capable of handling diverse types of time series data, including those collected regularly, irregularly, or continuously. In a more recent development, Zhou et al. (2020) introduced a comprehensive framework for continuous-time HMMs with an unknown number of states and transition patterns, designed to manage longitudinal data collected regularly, irregularly, or continuously. Their approach included a novel penalized method for estimating the number of hidden states in HMMs. Additionally, Luo and Stephens (2021) developed a reversible jump MCMC algorithm for continuous-time HMMs with an unknown number of states and clusters, implemented within a fully Bayesian framework. This model is adept at addressing challenges commonly encountered in latent multistate modeling, particularly irregular visits that vary among individuals.

In this paper, we employ the HMM to classify the electricity load curves of South Korean households. In addition, we explore in more detail how the number of hidden states in the HMMs, used to train each class, influences

classification performance. Traditionally, HMM-based classification employs the same number of hidden states for each class. For instance, Andersson (2017) employed identical hidden state numbers for HMM training, irrespective of whether load curves included a new DSM or not. However, this conventional approach may lead to suboptimal performance when the actual sets of hidden states from which the observation sequences originated differ across classes. In our study, we allowed for varying numbers of hidden states for each class, considering numerous potential sets. Subsequently, we compared the classification outcomes between HMMs employing the same number of hidden states for each class and those with different hidden state numbers. This comparative analysis enables us to assess the efficacy of employing class-specific hidden state numbers in enhancing the classification performance of the HMM.

The rest of the paper is organized as follows. Section 2 describes the data and how we assign classes on load curves for classification. Section 3 details how the HMM works and how it is trained. Descriptions of the HMM-based classification and model selection scheme are also provided. In Section 4, we present the classification results of the HMM models and compare them with various state-of-the-art models. Additionally, we analyze the load curves by making several inferences on the fitted HMM model. Section 5 concludes with a summary of key findings and their policy implications.

2 | DATA DESCRIPTION

Hourly electricity load data (recorded from January to December 2018) of several households were provided by the Korea Institute of Energy Technology Evaluation and Planning. Among these, 59 households of an apartment complex in Seoul, South Korea, were studied in this paper. A survey of these households was conducted on the number of home appliances a household owns based on type. The devices were divided into four categories: seasonal, video/audio, kitchen, and general, labeled A, B, C, and D, respectively, for convenience. Table 1 lists all devices asked about for each category.

TABLE 1 Devices discussed in the survey based on type.

Type	Devices included
Seasonal (A)	Air conditioner, humidifier, dehumidifier, electric blanket, electric fan
Video/Audio (B)	TV, computer monitor, audio device, DVD player
Kitchen (C)	Refrigerator, Kimchi refrigerator, electric rice cooker, electric oven Air fryer, dishwasher, microwave oven, induction range
General (D)	Air purifier, washing machine, tumble dryer, air dresser, vacuum cleaner Wireless vacuum cleaner, computer desktop, printer, boiler

The total number of devices for each type was converted into binary data, wherein class 1 denotes that the count is equal to or lower than a particular threshold, and class 2 denotes that the count is higher than the threshold. The threshold value was selected to divide the distribution as evenly as possible, resulting in four datasets for each device type with two classes for classification. Table 2 lists the mapping scheme of the two classes and their counts for each dataset.

Given the 1-year time period under consideration, which would be impractical to analyze using hourly data, the hourly electricity load data was aggregated into daily data to facilitate analysis. Figure 1 displays the average aggregated electricity

TABLE 2 Mapping scheme and counts of the classes.

Dataset	Class 1	Count	Class 2	Count
A	≤ 4	37	> 4	22
B	≤ 1	26	> 1	33
C	≤ 4	27	> 4	32
D	≤ 4	25	> 4	34

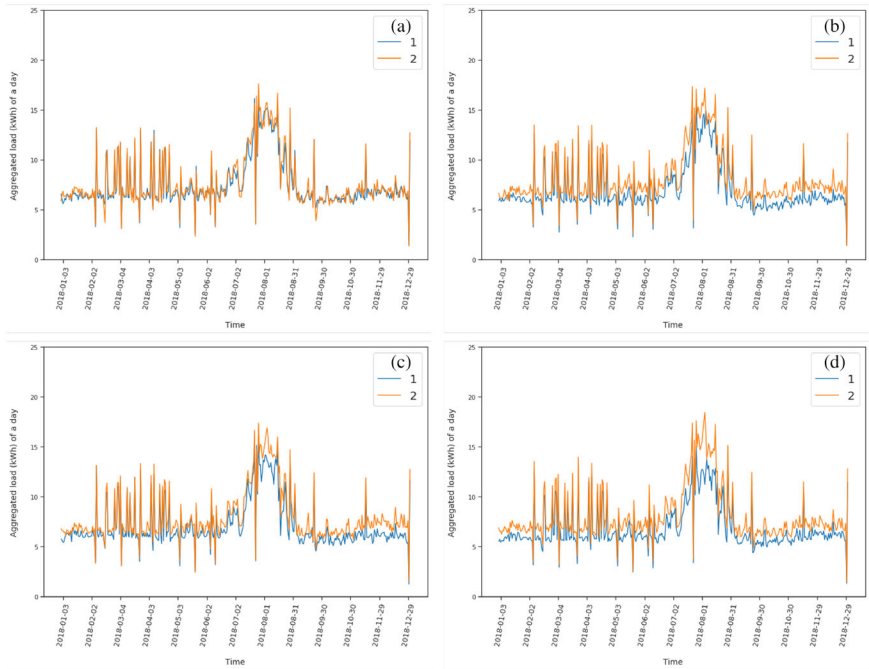


FIGURE 1 The average aggregated electricity load curves based on classes across multiple datasets. [Color figure can be viewed at [wileyonlinelibrary.com](https://onlinelibrary.wiley.com/terms-and-conditions)]

load curves for 59 households, categorized based on their classes using the mapping scheme for each dataset described in Table 2. It is evident that the load curve patterns vary depending on the number and type of devices that the households own. Of the four datasets, dataset A exhibits the smallest difference in averages between the two classes, while dataset D shows the largest difference.

3 | METHODS

3.1 | Hidden Markov model

HMM models a system in which a series of observations have come from a series of hidden states, which is assumed to be a Markov process. By definition, hidden states in the HMM are unobservable and must be learned from the observation sequence. We used the following notation by Jurafsky (2000) to describe an HMM. The observation sequence is defined as $O = \{o_1, o_2, \dots, o_T\}$, and the hidden state sequence is defined as $Q = \{q_1, q_2, \dots, q_T\}$. The possible values of each state are $S = \{s_1, s_2, \dots, s_N\}$ and the possible symbols per state are $V = \{v_1, v_2, \dots, v_M\}$. The parameters of HMM are $\lambda = \{A, B, \Pi\}$, where A denotes the transition probability matrix, B represents the emission probability matrix, and Π indicates the initial probability distribution. The transition probability matrix is defined as

$$A = \{a_{ij}\}, \sum_{j=1}^N a_{ij} = 1, i, j = 1, \dots, N,$$

where a_{ij} is the probability of moving from state i to state j . The emission probability matrix is defined as

$$B = \{b_i(v_k)\}, i = 1, \dots, N, k = 1, \dots, M,$$

where $b_i(v_k) = P(o_t = v_k | q_t = s_i)$, which is the probability of observing the symbol v_k given that the state at t is s_i . The initial probability distribution is

$$\Pi = \{\pi_i\}, i = 1, \dots, N,$$

where π_i indicates the probability that the Markov chain starts in state i and $\sum_{i=1}^N \pi_i = 1$.

The HMM can handle both discrete and continuous observation sequences. In the case of continuous data, the emission probability can assume a normal distribution. Then, the emission probability becomes $b_i(v_k) = N(v_k; \mu_i, \sigma_i^2)$ and

the parameters of the HMM become $\lambda = \{A, \mu, \sigma, \Pi\}$. Three tasks must be accomplished to use the HMM for analysis:

1. Computation of the likelihood of a particular observation sequence, given the model parameters, $P(O|\lambda)$, which can be achieved using the forward algorithm.
2. Decoding, a process of finding the most likely hidden state sequence in which a given observation sequence might have occurred from, $\operatorname{argmax}_Q P(Q|O, \lambda)$. The Viterbi algorithm is commonly used in this task.
3. Learning, finding the model parameters that best represent a given observation sequence, $\operatorname{argmax}_\lambda P(O|\lambda)$, which can be performed using the Baum-Welch algorithm.

Forward algorithm

Computation of the likelihood of a particular observation sequence given a model is crucial for classification using the HMM. Let $\alpha_t(j)$ be the forward probability representing the probability of being in state j after the first t observations given λ ,

$$\alpha_t(j) = P(o_1, \dots, o_t, q_t = j | \lambda).$$

Then, the value of forward probability, $\alpha_t(j)$, is calculated by summing the probabilities of all possible paths that lead to state j at time t ,

$$\alpha_t(j) = \sum_{i=1}^N a_{t-1}(j) a_{ij} b_j(o_t),$$

where $a_{t-1}(j)$ denotes the forward probability from the previous time step, a_{ij} indicates the probability of moving from state i to state j , and $b_j(v_k)$ represents the emission probability of v_k given the current state j . The forward algorithm described in [Algorithm 1](#) calculates the forward probabilities recursively and eventually obtains the likelihood of a particular observation sequence given the model parameters (Jurafsky, 2000).

Viterbi algorithm

The Viterbi algorithm is used to find the most likely sequence of hidden states where the observation sequence has come from. Let $v_t(j)$ be the Viterbi probability which is the probability that HMM is in state j after seeing the first

ALGORITHM 1 Forward algorithm

1. Initialization

$$\alpha_1(j) = \pi_j b_j(v_k), \quad 1 \leq j \leq N$$

2. Recursion

$$\alpha_t(j) = \sum_{i=1}^N \alpha_{t-1}(i) a_{ij} b_j(v_k), \quad 1 \leq j \leq N, \quad 2 \leq t \leq T$$

3. Result

$$P(O|\lambda) = \sum_{i=1}^N \alpha_T(i)$$

t observations and passing through the most likely hidden state sequence $(q_1, \dots, q_{t-1})$ given the parameters, λ . We can then express the Viterbi probability as

$$v_t(j) = \max_{q_1, \dots, q_{t-1}} P(o_1, \dots, o_t, q_1, \dots, q_{t-1}, q_t = j | \lambda).$$

If we already computed the probability of being in every state at time $t-1$, we can compute the Viterbi probability by taking the most probable path that leads to the current state at time t .

$$v_t(j) = \max_{i=1, \dots, N} v_{t-1}(i) a_{ij} b_j(o_t).$$

Algorithm 2 shows the formal structure of Viterbi algorithm. In order to trace the optimal sequence of hidden states, we need to keep track of the paths of hidden states that led to each state. We do this by storing the argument that maximizes $v_{t-1}(i) a_{ij} b_j(o_t)$ on ψ . P^* is the final probability obtained by the Viterbi algorithm and $Q^* = (q_1^*, \dots, q_T^*)$ is the predicted sequence of hidden states (Zucchini et al., 2017).

Baum-Welch algorithm

In many cases, we must estimate the parameters of the HMM based on an observation sequence. The Baum-Welch algorithm is a particular case of

ALGORITHM 2 Viterbi algorithm

1. Initialization

$$v_1(j) = \pi_j b_j(o_1), \quad 1 \leq j \leq N$$

$$\psi_1(j) = 0.$$

2. Recursion

$$v_t(j) = \max_{i=1, \dots, N} v_{t-1}(i) a_{ij} b_j(o_t), \quad 1 \leq j \leq N, \quad 2 \leq t \leq T$$

$$\psi_t(j) = \operatorname{argmax}_{i=1, \dots, N} v_{t-1}(i) a_{ij} b_j(o_t), \quad 1 \leq j \leq N, \quad 2 \leq t \leq T.$$

3. Termination

$$P^* = \max_{i=1, \dots, N} v_T(i),$$

$$q_T^* = \operatorname{argmax}_{i=1, \dots, N} v_T(i).$$

4. Result (Backtracing)

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, \dots, 1.$$

an iterative algorithm called the expectation–maximization (EM) algorithm (Jurafsky, 2000), which computes initial estimates for the transition and emission probabilities and uses them to derive better estimates iteratively.

Backward probability is introduced to estimate transition probabilities, A , emission probabilities, B , and initial probability distribution, Π . Backward probability, $\beta_t(i)$ is the probability of observations from time $t+1$ to T , given that the state is i at time t with a model whose parameters are $\lambda = (A, B, \Pi)$,

$$\beta_t(i) = P(o_{t+1}, o_{t+2}, \dots, o_T | q_t = i, \lambda).$$

It is calculated recursively in a manner similar to the forward algorithm. The transition probabilities, A , are estimated as follows:

$$\hat{a}_{ij} = \frac{\text{expected frequency of transitions from state } i \text{ to state } j}{\text{expected frequency of transitions from state } i}$$

$$= \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \sum_{k=1}^N \xi_t(i, k)},$$

where $\xi_t(i, j)$ denotes the probability of being in state i at time t and state j at time $t + 1$, given the observation sequence and model parameters:

$$\begin{aligned}\xi_t(i, j) &= P(q_t = i, q_{t+1} = j | O, \lambda) \\ &= \frac{P(q_t = i, q_{t+1} = j, O | \lambda)}{P(O | \lambda)} \\ &= \frac{\alpha_t(j) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{j=1}^N \alpha_t(j) \beta_t(j)}.\end{aligned}$$

The estimates for emission probabilities B are computed as follows:

$$\begin{aligned}\hat{b}_j(v_k) &= \frac{\text{expected frequency of being in state } j \text{ and observing } v_k}{\text{expected frequency of being in state } j} \\ &= \frac{\sum_{\{t: o_t = v_k\}} \gamma_t(j)}{\sum_{t=1}^T \gamma_t(j)},\end{aligned}$$

where $\gamma_t(j)$ denotes the probability of being in state j at time t ,

$$\begin{aligned}\gamma_t(j) &= P(q_t = j | O, \lambda) \\ &= \frac{P(q_t = j, O | \lambda)}{P(O | \lambda)} \\ &= \frac{\alpha_t(j) \beta_t(j)}{P(O | \lambda)}.\end{aligned}$$

Last, the initial probability distribution is estimated as

$$\hat{\pi}_i = \gamma_1(i),$$

where $\gamma_1(i)$ is the expected frequency of being in state i at time $t = 1$.

With the above calculations, we can now reestimate $\lambda = (A, B, \Pi)$ with their previous estimations. The Baum-Welch algorithm based on the EM algorithm is described in [Algorithm 3](#).

3.2 | HMM-based classification

Classification of time series using HMM involves training the parameters of the HMM and computing the likelihood of a given observation sequence (Zucchini

ALGORITHM 3 Baum-Welch algorithm

1. Initialization

$$\text{Initial } \lambda = (A, B, \Pi)$$

2. Iteration

Expectation step

$$\text{Compute } \gamma_i(j) \text{ and } \xi_i(i, j) \text{ with } (A, B, \Pi)$$

Maximization step

$$\text{Compute } \hat{a}_{ij}, \hat{b}_j(v_k), \text{ and } \hat{\pi}_j \text{ with } \gamma_i(j) \text{ and } \xi_i(i, j)$$

Update

$$(A, B, \Pi) = (\hat{A}, \hat{B}, \hat{\Pi})$$

3. Result

$$\text{Converged } \lambda = (A, B, \Pi)$$

et al., 2017). First, separate HMMs are trained for each class of a dataset using the Baum-Welch algorithm:

$$\lambda_c = (A_c, B_c, \Pi_c), \quad c = 1, \dots, C,$$

where C denotes the number of classes. Then, we compute the likelihood of an observation sequence with each model using the forward algorithm:

$$P(O|\lambda_c), \quad c = 1, \dots, C.$$

Finally, the observation sequence is classified into a class of the model that provided the highest likelihood:

$$k = \underset{c}{\operatorname{argmax}} P(O|\lambda_c).$$

The HMMs in this study were implemented using the Python package **RIGHTSLINK** (v, 2016).

3.3 | Model selection

One of the critical hyperparameters of the HMM is the number of hidden states specified before learning the parameters $\lambda = (A, B, \Pi)$. Commonly used model selection criteria such as the AIC and BIC, are used to select the optimal number of hidden states for forecasting using the HMM (Dofadar et al., 2021). For classification, a grid search can be used to determine the optimal number of hidden states that maximizes the classification metrics. The maximum number of hidden states need not be high because such models are difficult to interpret, and many limit the number to a single digit (Andersson, 2017). In this study, we consider the number of possible hidden states N in the range of two to nine. The optimal number of hidden states of the HMM was chosen via grid search that maximizes the classification performance.

Generally, HMM-based classification assigns HMMs with the same number of hidden states to each class to train with. For example, Andersson (2017) used the same number of hidden states of the HMM to train the load curves with and without new DSM. However, if the actual sets of hidden states from which the observation sequences have originated are different for each class, this method could yield poor results. In this study, we investigate the effectiveness of using a class-specific number of hidden states by training the HMM with different numbers of hidden states for each class. This approach enables the HMM to capture information that is specific to each class, thus improving classification performance. To find the optimal number of hidden states for each class, we consider all possible combinations in the range of two to nine and select the set $(N_1, \dots, N_C)$ that yields the best classification performance using grid search. This model is referred to as HMM* in Section 4 to differentiate it from the standard HMM.

3.4 | Evaluation metrics

We evaluate the classification performance of our models by considering both accuracy and weighted F1 score as evaluation metrics. For binary classification problems, a confusion matrix can be used to evaluate the performance of a classifier. If we define class 1 as the negative class and class 2 as the positive class, then the true positive (TP) represents the number of samples correctly predicted as positive, false positive (FP) is the number of samples incorrectly predicted as positive, true negative (TN) is the number of samples correctly predicted as negative, and false negative (FN) is the number of samples incorrectly predicted as negative. We can calculate accuracy as

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}},$$

and the F1 score as

$$\text{F1 score} = \frac{\text{TP}}{\text{TP} + \frac{1}{2}(\text{FP} + \text{FN})}.$$

The weighted F1 score is calculated as the weighted average of the F1 scores for each class, where the weights are determined by the relative frequency of each class in the dataset. The weighted F1 score is considered an ideal metric for imbalanced datasets since it incorporates the support of each class.

4 | DATA ANALYSIS

In this section, we have classified the load curves based on the device types described in Section 2. To accomplish HMM-based classification, we have utilized two models: standard HMM, which has been trained with the same number of hidden states for each class, and HMM*, which has been trained with different numbers of hidden states for each class. We compared the classification performance of HMM-based classification with various state-of-the-art models. For the distance-based method, we employed KNN algorithm with DTW metric. For feature-based method, we used BOSSVS, and bag-of-features approach (TSBF) which were implemented by the python package pyts (Faouzi & Janati, 2020). For deep learning models, we used the FCN and its variants using LSTM-FCN and GRU-FCN, IT, and OS-CNN which were implemented using PyTorch (Paszke et al., 2019), as well as more recent methods, MIN-IROCKET and TST. Furthermore, we provide inferences made on the fitted HMM model to obtain deeper insights into the load curves.

4.1 | Classification

We first demonstrate how HMM-based classification classifies an observation sequence to its class based on the likelihood computation. In this study, we computed the log of likelihood as follows:

$$LL_c = \log[P(O|\lambda_c)], \quad c = 1, 2,$$

TABLE 3 Classification of dataset A by HMM (with the same number of hidden states for each class).

Household ID	True	LL_1	LL_2	Predicted
0	2	-1026.29	-1036.01	1
7	1	-928.45	-1034.96	1
18	1	-1077.24	-1126.43	1
21	1	-844.00	-861.48	1
22	1	-1093.37	-1142.65	1
24	2	-842.89	-827.93	2
33	1	-903.89	-905.11	1
35	1	-850.87	-843.92	2

Note: The larger log-likelihood for each household is in bold.

TABLE 4 Classification of dataset A by HMM* (with different numbers of hidden states for each class).

Household ID	True	LL_1	LL_2	Predicted
0	2	-1026.29	-1010.02	2
7	1	-928.45	-1038.98	1
18	1	-1077.24	-1157.01	1
21	1	-844.00	-892.70	1
22	1	-1093.37	-1261.15	1
24	2	-842.89	-827.83	2
33	1	-903.89	-918.07	1
35	1	-850.87	-871.05	1

Note: The larger log-likelihood for each household is in bold.

where c indicates the class. We computed the log-likelihood values for each observation sequence of a household in a given test set, using models that were trained on class 1 and class 2, respectively. We subsequently assigned the household to the class with the larger likelihood value. Table 3 presents the classification result of the second fold of the 8-fold cross-validation for dataset A using HMM. The number of hidden states was three, and it achieved an accuracy of 0.750 and weighted F1 score of 0.750.

We further show how HMM* was trained with three hidden states for class 1 and two hidden states for class 2 performed the classification. Table 4 presents the classification result of HMM* which indicates the classes were correctly predicted for all households in the dataset. It is evident that enabling the model

TABLE 5 Averages of the weighted F1 scores and accuracies, respectively, obtained by each model through the eight-fold cross-validation.

Dataset	HMM*	HMM	KNN	BOSSVS	TSBF	FCN	LSTM-FCN	GRU-FCN	IT	OS-CNN	MINIROCKET	TST
(a) Weighted F1 scores												
A	0.781 (1)	0.593 (2)	0.580 (3)	0.444 (8)	0.431 (9)	0.498 (4)	0.469 (6)	0.452 (7)	0.495 (5)	0.392 (11)	0.366 (12)	0.419 (10)
B	0.840 (1)	0.805 (2)	0.526 (8)	0.657 (3)	0.458 (11)	0.425 (12)	0.654 (4)	0.549 (7)	0.492 (10)	0.550 (6)	0.579 (5)	0.514 (9)
C	0.737 (1)	0.672 (2)	0.533 (5)	0.584 (3)	0.521 (6)	0.396 (12)	0.424 (10)	0.462 (7)	0.457 (8)	0.443 (9)	0.542 (4)	0.408 (11)
D	0.864 (1)	0.775 (2)	0.429 (12)	0.625 (3)	0.530 (8)	0.518 (9)	0.571 (6)	0.562 (7)	0.621 (4)	0.573 (5)	0.514 (11)	0.516 (10)
Average rank	1 (1)	2 (2)	7 (6)	4.25 (3)	8.5 (10)	9.25 (11)	6.5 (4)	7 (6)	6.75 (5)	7.75 (8)	8 (9)	10 (12)
(b) Accuracies												
A	0.795 (1)	0.589 (4)	0.609 (3)	0.438 (9)	0.458 (8)	0.627 (2)	0.509 (6)	0.478 (7)	0.558 (5)	0.424 (10)	0.375 (12)	0.413 (11)
B	0.848 (1)	0.812 (2)	0.531 (9)	0.647 (4)	0.462 (12)	0.560 (6)	0.649 (3)	0.544 (7)	0.511 (11)	0.540 (8)	0.601 (5)	0.518 (10)
C	0.746 (1)	0.676 (2)	0.529 (5)	0.596 (3)	0.522 (7)	0.544 (4)	0.476 (9)	0.507 (8)	0.464 (10)	0.444 (11)	0.524 (6)	0.425 (12)
D	0.866 (1)	0.766 (2)	0.438 (12)	0.612 (5)	0.547 (10)	0.645 (3)	0.582 (6)	0.562 (8)	0.634 (4)	0.576 (7)	0.548 (9)	0.528 (11)
Average rank	1 (1)	2.5 (2)	7.25 (6)	5.25 (4)	9.25 (11)	3.75 (3)	6 (5)	7.5 (7)	7.5 (7)	9 (10)	8 (9)	11 (12)

Note: The relative ranking of each model is given in parentheses.

to train using a different number of hidden states for class 2 has led to an improvement in the classification result.

Table 5 shows the average values of the weighted F1 scores and accuracies, respectively obtained by the models through the eight-fold cross-validation. The results demonstrate that the HMM models outperformed other competing models across all datasets in terms of both weighted F1 score and accuracy. Although the distance-based model (KNN) and feature-based model (BOSSVS) showed relatively good results compared to other models, they still did not perform as well as the HMM models. In addition, the HMM* achieved even better results than the HMM, indicating that using class-specific number of hidden states can improve the classification performance of the HMM.

Considering that the datasets examined in this study have small sample sizes, it is evident that the HMM is able to achieve better performance on the classification of time series datasets with small sample sizes. This elucidates the relatively poorer results achieved by many deep learning models, as these models are recognized for their need for larger sample sizes to facilitate effective training. It is noteworthy that some models exhibited fluctuating rankings depending on the evaluation metric employed, either weighted F1 score or accuracy, while the HMM models sustained consistent rankings across both metrics. Given that some of the test sets were imbalanced, this implies that HMM-based classification is more robust to class imbalance than other competing models.

4.2 | Inference

A notable advantage of employing HMM-based classification lies in the ability to perform inferences on the models that are trained for each individual class of the time series. Table 6 shows the parameter estimates of the Gaussian HMM trained on dataset A (seasonal devices) with three hidden states for both classes. We intentionally chose to make inferences on the HMM, because we found that, although the classification performance was superior with two hidden states for

TABLE 6 Parameters of the Gaussian HMMs trained on each class of dataset A.

Model	State	Mean	SD
Class 1	1	5.171	1.680
	2	8.507	2.460
	3	10.973	7.590
Class 2	1	5.143	1.296
	2	8.109	2.426
	3	12.192	6.161

class 2 based on the HMM* result, it was deemed more straightforward to interpret by maintaining consistency in the number of hidden states between the two classes.

The hidden states in HMM are intrinsically unobservable, and their actual meaning is often open to the researcher's discretion for interpretation, relying on the observed data and model parameters (Zucchini et al., 2017). Since we assume that there are three states where the load curves were theoretically generated, it may be appropriate to interpret the states as periods of low, medium, and high energy demand by a household.

Table 6 represents that the estimated means of states 1 and 2 for the HMM model trained on class 1 are larger than those for the model trained on class 2, while the mean of state 3 is larger for the model trained on class 2 than the one trained on class 1. This implies that households with fewer electronic devices tend to have higher energy demands on average during periods of low and medium energy demand, while those with more electronic devices tend to have higher energy demands on average during periods of high energy demand. The large standard deviations of state 3 for both classes indicate significant variability in energy demand during periods of high demand, irrespective of the number of electronic devices in a household.

Figure 2 illustrates the transition probability matrices of the models trained on each class. Both matrices indicate that there are relatively few transitions between states. However, the model trained in class 2 tends to move more frequently from state 1 to state 3 and from state 2 to state 3 than the model trained in class 1. This is consistent with our expectations, as households with many electronic devices are more likely to move from periods of low or moderate

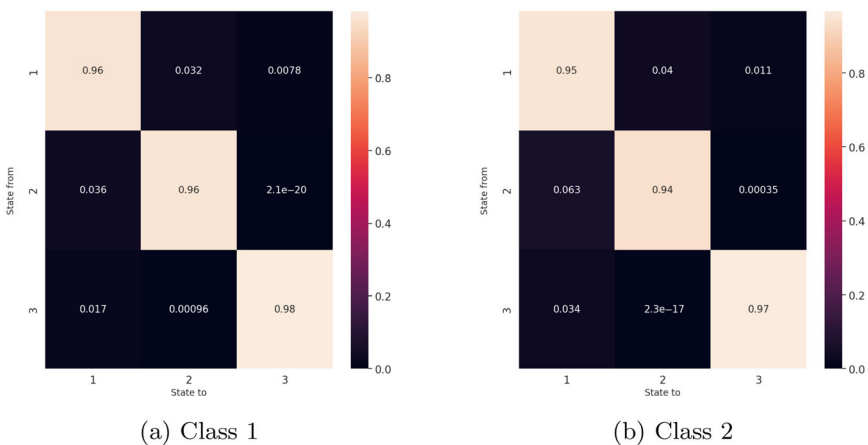
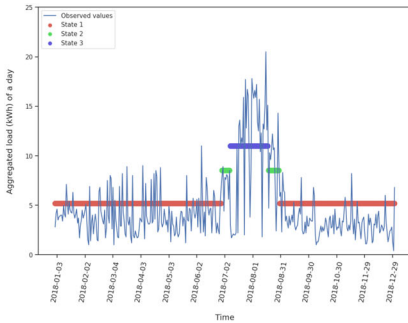
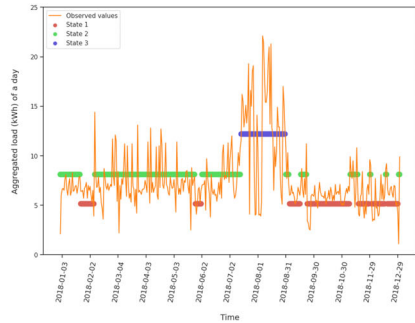


FIGURE 2 Transition probability matrices of the HMMs trained on each class of dataset A. [Color figure can be viewed at [wileyonlinelibrary.com](https://onlinelibrary.wiley.com/doi/10.1111/asa1.12338)]



(a) Class 1



(b) Class 2

FIGURE 3 Time series plots of samples from each class of dataset A with the means of the most likely states. [Color figure can be viewed at wileyonlinelibrary.com]

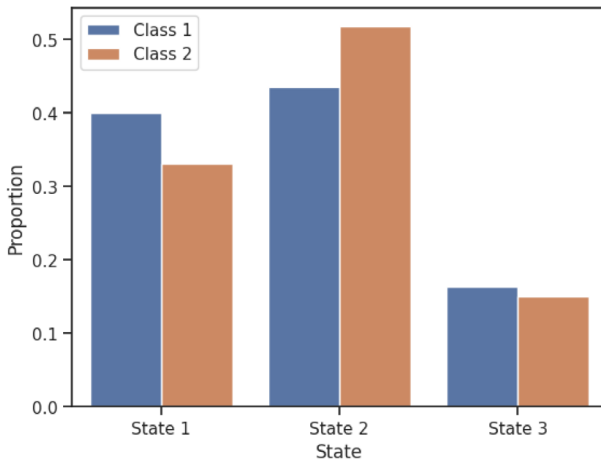


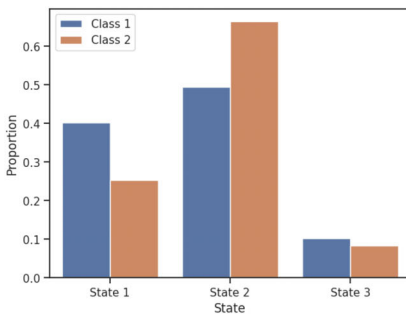
FIGURE 4 Proportions of the most likely states of dataset A for each class. [Color figure can be viewed at wileyonlinelibrary.com]

energy demand to periods of high energy demand compared to those with fewer electronic devices.

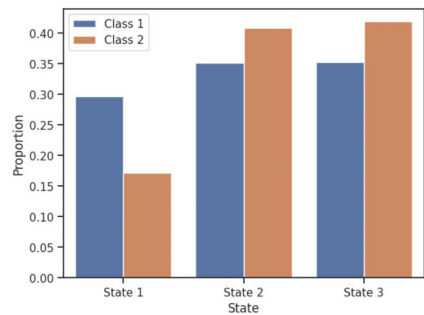
Figure 3 displays time series plots of sample households from both classes of dataset A, with the means of the most likely states predicted by the Viterbi algorithm plotted alongside. The plots clearly illustrate that the household from class 1 spends more time in state 1, which is a period of low energy demand, while the household from class 2 spends more time in state 2, which is a period of medium energy demand. This result makes sense since households with fewer electronic devices are expected to have less energy demand and thus spend more

time in low energy states, whereas households with many electronic devices tend to have more energy demand and spend more time in medium energy states. Furthermore, both households show a tendency to stay in state 3, which is a period of high energy demand throughout the summer season.

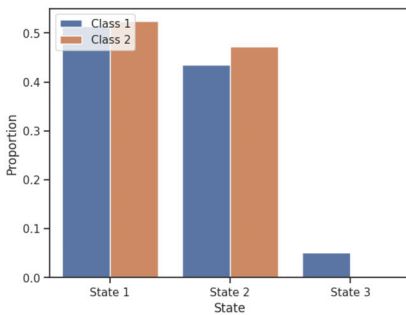
The proportions of the most likely states predicted by the Viterbi algorithm for each class of dataset A are presented in Figure 4. The figure shows that the proportion of state 1 is higher in class 1 than in class 2, while the proportion of state 2 is higher in class 2 than in class 1. Both classes have comparable proportions of state 3, although class 1 has a slightly higher proportion. These findings are consistent with the interpretation of the three states as denoting periods of low, medium, and high energy demand, because households with few electronic devices are more likely to have longer periods of low energy demand, while households with many electronic devices are more likely to have longer periods of medium energy demand. In terms of economic policy, policymakers can promote energy efficiency programs and provide subsidies to households with few



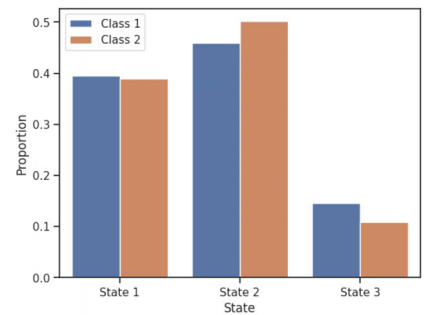
(a) Spring



(b) Summer



(c) Autumn



(d) Winter

FIGURE 5 Proportions of the most likely states of dataset A for each class based on four seasons. [Color figure can be viewed at [wileyonlinelibrary.com](https://onlinelibrary.wiley.com/doi/10.1111/asa.12338)]

electronic devices to maintain their low energy consumption, while encouraging the use of smart home systems and offering incentives for energy-efficient upgrades to reduce medium energy demand in households with many electronic devices.

Figure 5 provides additional insights into the energy demand patterns by showing the proportions of the states based on seasons. In general, the seasonal proportions are consistent with those in Figure 4, with two notable exceptions: summer and autumn. During summer, the proportions of state 3 increased for both classes, but the increase was more pronounced for class 2, which has significantly larger proportions of state 3 than class 1. This implies that the energy demand escalates substantially for both classes during the summer season, but the rise is even more pronounced for households with a larger number of electronic devices. In autumn, the proportion of state 1 increased dramatically for both classes, while the proportion of state 3 decreased. The decrease of state 3 was more pronounced for class 2, suggesting that the energy demand is significantly lower in autumn for both classes compared to other seasons, particularly for households with many electronic devices. These findings provide valuable insights for developing economic policies and energy management strategies that account for seasonal variations in energy demand. By tailoring policies to address the specific needs and behaviors observed in different seasons, policymakers can enhance energy efficiency, ensure grid stability, and promote sustainable energy use.

5 | CONCLUDING REMARKS

Understanding energy demand patterns in the residential sector is crucial for optimizing energy supply and consumption, which has significant implications for economic policy and resource management. To make a contribution to this endeavor, we conducted load curve classification to gain insight into the energy demand behavior of households residing in an apartment complex located in South Korea. Our approach involved labeling load curves for each household based on home appliance types and their counts. We employed an HMM-based classification and investigated the effect of class-specific hidden state numbers on the classification performance of the HMM. To evaluate the performance of HMM-based classification, we compared it to various state-of-the-art time series classification models, including distance-based, feature-based, and deep learning methods.

Our results demonstrate that the HMM models consistently outperformed other competing models across all datasets. Notably, HMM* with class-specific numbers of hidden states achieved better performance than HMM with the same number of hidden states for each class. This suggests that allowing different numbers of hidden states for each class can improve the classification

performance of the HMM. Additionally, our findings indicate that the HMM models can effectively handle small or imbalanced datasets, as evidenced by high scores on evaluation metrics such as weighted F1 score and accuracy.

To obtain deeper insights into the load curves, we performed inference on the fitted HMM. As an illustrative case, we employed the HMM trained with three hidden states for both classes on dataset A. The states were interpreted as denoting periods of low, medium, and high energy demand. We ascertained that this interpretation was in line with the parameters of the fitted HMM. Our analysis revealed that households with fewer electronic devices tend to have longer periods of low energy demand, while households with more electronic devices have longer periods of medium energy demand. These findings suggest several economic and policy implications for managing residential energy demand. Targeted energy efficiency programs can focus on households with few electronic devices by promoting the adoption of energy-efficient appliances through subsidies, maintaining their low consumption levels. For households with many devices, policies could encourage the use of smart home systems and offer incentives for upgrading to energy-efficient appliances to reduce medium demand periods. DSM strategies, such as dynamic pricing, can help shift energy use to lower demand periods. Infrastructure planning should account for the higher loads from households with many devices, ensuring grid stability and integrating renewable energy sources to match high demand periods. Economic incentives like appliance rebates and smart home technology adoption can further aid in managing energy consumption efficiently, promoting overall energy savings and sustainability.

We also noted that both groups demonstrated a rise in the proportion of medium and high energy demand periods during the summer season while having a larger proportion of low energy demand periods during autumn. These results indicate that seasonal energy pricing policies, such as implementing higher tariffs during peak summer months to reduce high energy demand and offering autumn discounts to balance grid load, can manage seasonal fluctuations. Energy efficiency incentives like rebates for energy-efficient cooling systems in summer and upgrades during the lower demand in autumn can enhance overall efficiency. Additionally, integrating renewable energy sources, such as supporting solar panel installations in summer and wind energy generation in autumn, can help meet seasonal energy demands sustainably.

Although the current datasets used in the analysis showed promising results in classifying load curves using HMM-based classification, more extensive datasets should be considered in future work to assess the performance of this method more thoroughly. In addition, various variants of the HMM have been successful in modeling load curves, including hierarchical HMM and factorial HMM, which can handle data of multiple levels. Since the survey conducted for this study also provides information regarding the size of each residence, it may be more appropriate to model the load curves using methods that can

accommodate multiple levels, particularly because larger houses tend to exhibit higher energy consumption. In future work, we plan to utilize these HMM variants to gain further insight into the energy demand behavior of the households from the perspective of economic policy and energy management.

ACKNOWLEDGMENTS

This research was supported by the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (NRF-2019R1C1C1011710) and supported by the Korea Institute of Energy Technology Evaluation and Planning (KETEP) and the Ministry of Trade, Industry & Energy (MOTIE) of the Republic of Korea (No. 20199710100060).

DATA AVAILABILITY STATEMENT

Some or all data, models, or codes that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Beom Seuk Hwang  <https://orcid.org/0000-0001-7224-1127>

REFERENCES

- Andersson, M. (2017). Modeling electricity load curves with hidden Markov models for demand-side management status estimation. *International Transactions on Electrical Energy Systems*, 27(3), e2265.
- Axelrod, S., & Maison, B. (2004). Combination of hidden Markov models with dynamic time warping for speech recognition. *2004 IEEE ICASSP*, Montreal, QC, Canada, 1-173.
- Balasubramanian, S., & Balachandra, P. (2021). Characterising electricity demand through load curve clustering: A case of Karnataka electricity system in India. *Computers & Chemical Engineering*, 150, 107316.
- Baydogan, M. G., Runger, G., & Tuv, E. (2013). A bag-of-features framework to classify time series. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11), 2796-2802.
- Christ, M., Braun, N., Neuffer, J., & Kempa-Liehr, A. W. (2018). Time series feature extraction on basis of scalable hypothesis tests (tsfresh—A python package). *Neurocomputing*, 307, 72-77.
- Dempster, A., Petitjean, F., & Webb, G. I. (2020). ROCKET: Exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery*, 34(5), 1454-1495.
- Dofadar, D. F., Khan, R. H., & Alam, M. G. R. (2021). COVID-19 confirmed cases and deaths prediction in Bangladesh using hidden Markov model. *4th International Conference on BioSMART*, 1-4.
- Elsayed, N., Maida, A. S., & Bayoumi, M. (2019). Gated recurrent neural networks empirical utilization for time series classification. In *2019 International Conference on iThings and IEEE GreenCom and IEEE CPSCom and IEEE Smart Data*, 1207-1210.
- Faouzi, J., & Janati, H. (2020). pyts: A python package for time series classification. *The Journal of Machine Learning Research*, 21(1), 1720-1725.
- Firth, S., Lomas, K., Wright, A., & Wall, R. (2008). Identifying trends in the use of domestic appliances from household electricity consumption measurements. *Energy and Buildings*, 40(5), 926-936.

- Guédon, Y. (2003). Estimating hidden semi-Markov chains from discrete sequences. *Journal of Computational and Graphical Statistics*, 12(3), 604–639.
- Hu, S., Yan, D., & Qian, M. (2019). Using bottom-up model to analyze cooling energy consumption in China's urban residential building. *Energy and Buildings*, 202, 109352.
- Ismaïl Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., & Muller, P. A. (2019). Deep learning for time series classification: A review. *Data Mining and Knowledge Discovery*, 33(4), 917–963.
- Jurafsky, D. (2000). *Speech & language processing*. Pearson Education India.
- Karim, F., Majumdar, S., Darabi, H., & Chen, S. (2017). LSTM fully convolutional networks for time series classification. *IEEE Access*, 6, 1662–1669.
- Lebedev, S. (2016). hmmlearn. GitHub. <https://github.com/hmmlearn/hmmlearn>
- Luo, Y., & Stephens, D. A. (2021). Bayesian inference for continuous-time hidden Markov models with an unknown number of states. *Statistics and Computing*, 31, 57.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., & Desmaison, A. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32. https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=pytorch%3A+an+imperative+style%2C+high-performance+deep+learning+library&btnG=
- Premkumar, R., Shanmugasundaram, R., Thilak, K. R., & Balaji, A. N. (2022). A review on demand side management: Definition, scope, challenges and benefits. 8th ICSSS, 1–4.
- Schäfer, P. (2015). Bag-Of-SFA-Symbols in Vector Space (BOSS VS). ZIB report, 15–30.
- Tan, C. W., Petitjean, F., & Webb, G. (2020). FastEE: Fast ensembles of elastic distances for time series classification. *Data Mining and Knowledge Discovery*, 34, 1–42.
- Tang, W., Long, G., Liu, L., Zhou, T., Blumenstein, M., & Jiang, J. (2020). Omni-scale CNNs: A simple and effective kernel size configuration for time series classification. *arXiv preprint arXiv:2002.10061*.
- Wang, Z., Yan, W., & Oates, T. (2017). Time series classification from scratch with deep neural networks: A strong baseline. In *International joint conference on neural networks (IJCNN)* (pp. 1578–1585). IEEE.
- Zerveas, G., Jayaraman, S., Patel, D., Bhamidipaty, A., & Eickhoff, C. (2021). A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2114–2124.
- Zhou, J., Song, X., & Sun, L. (2020). Continuous time hidden Markov model for longitudinal data. *Journal of Multivariate Analysis*, 179, 104646.
- Zucchini, W., MacDonald, I. L., & Langrock, R. (2017). *Hidden Markov models for time series: An introduction using R*. CRC Press.

How to cite this article: Lee, J., & Hwang, B. S. (2024). Energy demand pattern analysis in South Korea using hidden Markov model-based classification. *Asian Economic Journal*, 38(3), 404–428. <https://doi.org/10.1111/asej.12338>