IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 15 December 2022, accepted 3 January 2023, date of publication 10 January 2023, date of current version 19 January 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3235724

RESEARCH ARTICLE

Spatiotemporal Sequence-to-Sequence Clustering for Electric Load Forecasting

MOSES AMOASI ACQUAH[1], (Member, IEEE), YUWEI JIN[2], (Member, IEEE), BYEONG-CHAN OH[1], (Graduate Student Member, IEEE), YEONG-GEON SON[1], (Graduate Student Member, IEEE), AND SUNG-YUL KIM[1], (Member, IEEE)

[1]Department of Electrical Energy Engineering, Keimyung University, Daegu 42601, South Korea
[2]Department of Electrical and Electronic Engineering, Kyungpook National University, Daegu 41566, South Korea

Corresponding author: Sung-Yul Kim (pslab2040@gmail.com)

This work was supported in part by the Project Open Innovation Research and Development under Grant 21-CP-002, and in part by K-water.

ABSTRACT Massive electrical load exhibits many patterns making it difficult for forecast algorithms to generalise well. Most learning algorithms produce a better forecast for dominant patterns in the case of weekday consumption and otherwise for less dominant patterns in weekend and holiday consumption. In view of this, there is the need to cluster the load patterns, so learning algorithms can focus on the patterns independently to produce forecasts with better accuracy for all cases. However, clustering time-series data breaks the time-series dependency, making model training difficult. This paper presents a novel sequence-to-sequence cluster framework to reform time-series dependency after clustering; this enables independent clusters to be modelled using Convolutional Neural Network-Gated Recurrent Unit, which learns spatiotemporal features for future forecasts. A real-world dataset by the Korea Power Exchange composed of nationwide consumption is used for case studies and experiments. Experimental results verify that the proposed study effectively improves the accuracy of electric load forecasting by about 50%, with a WAPE of 0.67%. The proposed method also speeds up the training process of the forecast algorithm by about 35%, given that only a subset of the dataset is trained due to clustering. Korea Water Resources Corporation has implemented the proposed method for load forecasting and system marginal price estimation.

INDEX TERMS Convolutional neural network-gated recurrent unit (CNN-GRU), feature engineering, k-means clustering, LightGBM classifier, sequence-to-sequence forecast, short-term load forecast (STLF).

I. INTRODUCTION

Rapid technological advancements have accelerated energy consumption in buildings across all walks of life [1]. Modern building energy systems integrate resources, such as electric vehicles (EVs) and heating, ventilating, and air conditioning (HVAC), for smart grid scheduling [1]. With this growth comes the need to increase energy production and introduce different energy mixes into the grid; this is a concern owing to the high emission of greenhouse gases associated with electricity production, inefficiencies related to energy consumption, and high tariff on energy consumption [2]. Energy generation and consumption planning need to be enforced to mitigate these problems. This can be achieved through accurate load forecasting.

Load forecasting can also assist in the scheduling system marginal price (SMP) for energy market optimisation. The SMP price forecast is vital for optimal bidding on the energy market for market stabilisation and economic benefit. Since load power demand and SMP are closely related, an accurate load forecast must be considered to determine the exact hourly SMP [3].

To this end, much research has been conducted on electrical load forecast for economic benefit, efficient operation, and building energy management.

Load forecasting can be categorised into statistical-based forecasting and artificial intelligent-based forecasting [4].

The associate editor coordinating the review of this manuscript and approving it for publication was Zhigao Zheng.

Statistical-based forecasting applies mathematical theory to model time series patterns. Examples include Linear Regression (LR), Support Vector Machines (SVM), and Autoregressive Integrated Moving Average (ARIMA) models [5], [6], [7]. These methods and their variant employ powerful time-series techniques that are simple to use with high computation speed. Nonetheless, since these methods are bent on the stationarity of time series, they have very low accuracy when forecasting electrical loads with high non-linear features.

Comparatively, neural network-based artificial intelligence models mimic the human brain and have shown impressive results in learning important details from massive power load data. There are various types of Artificial Intelligence (AI) forecasting models, aka machine learning techniques. References [8] and [9] proposed a Back Propagation Neural Network (BPNN) power load forecasting for electricity grids with reasonable accuracy. However, BPNN suffers from overfitting and does not generalise well. By default, they can also not capture non-linear hidden features such as time dependency, trends, and seasonality. Due to this, feature engineering that captures time-series trends and seasonality needs to be integrated [10].

Over the years, Recurrent Neural Network (RNN) has proved effective and suitable for time-series load forecasting as it is capable of non-linear problems. Reference [11] discusses household load forecasting via an RNN. However, it suffers from a diminishing gradient, making it forgetful and falling short in practical application.

Long short-term memory (LSTM) is an improved RNN that considers long-term correlations in a dataset. References [12] and [13] applies LSTM to better electric load forecasting accuracy. Although the LSTM networks are better at dealing with time-series data, one main drawback is that they capture temporal variations in sequential data while ignoring correlations between the input features.

A Gated recurrent unit (GRU) network mitigates the shortfall of LSTM networks. Reference [14] proposes a GRU model to forecast energy consumption. Comparatively, GRU architecture is simple and has fewer parameters with a faster convergence time than LSTM. However, neither LSTM nor GRU is capable of analysing discontinuous data. The study by [15] and [16] employed an ensemble LSTM hybrid model to forecast electric load accurately. The comparative analysis validated the efficiency of the proposed method.

Convolutional Neural Network (CNN) is mostly used to extract special features from a dataset [17]. Spatial features provide contextual information and a visual appearance of a section of the load profile (pattern), while temporal characteristics define the relative correlation between load records [18]. In such a case, the one-dimensional CNN shows unique advantages of extracting non-linear features from electric load data. Many researchers resort to CNN and LSTM/GRU to extract spatiotemporal features for load forecasting. The works in [19] and [20] employed CNN-LSTM and CNN-GRU hybrid models for load forecasting; comparative results show higher accuracy than electric load forecasting with non-hybrid models.

Though CNN-GRU shows relatively promising results in accuracy, its training process is time-consuming; also, when the load data exhibit high non-linearity, such as multiple seasonality, its accuracy reduces [21]. To mitigate the issues above, most researchers introduce clustering to extract and separate non-linear patterns in load consumption for accurate forecasting. Reference [22] proposes load forecasting for individual users of a building. This method uses k-means to cluster a load of multiple users into similar groups. A Back Propagation Neural Network (BPNN) is utilised for short-term load forecasting. Reference [23] developed a Pyramid-CNN model for feature extraction by taking advantage of its convolutional layers; also, the pyramidal architecture allows for complexity reduction; the DBSCAN algorithm is employed to group similar electricity users and provide group predictions. The work in [10] proposed a load forecasting framework employing clustering for pattern extraction and a classification method to label the data patterns for future forecasts. Clustered labels with historical data generate and encode categorical features for future forecasts. Reference [24] presented K-means clustering with CNN network for load forecast, big data is clustered into sub-datasets to train a CNN model to improve forecast accuracy.

As big data exhibit different patterns of variable proportions, the future forecast does not generalise well in some cases. In the case of electric load, weekday forecasts are primarily good while weekends and holidays fail woefully [25]; a way around this is a process of tuning, which takes much time and cannot guarantee success [26]. Weekend and holiday load forecasts for planning energy management systems are equally important; when scheduling SMP for energy market optimisation, significant forecast errors lead to high economic penalties [3]. To mitigate this problem, it is prudent to isolate possible similar partners in a dataset as a cluster and use the individual clusters to learn futures forecasts independently. The problem with this approach is that the cluster a future forecast belongs to needs to be determined beforehand. Reference [10] proposed an early classification method to determine future forecast horizons ahead of time using the Light Gradient Boosting Machine algorithm (LightGBM or LGBM), but the forecast model uses this information for feature extraction and not direct forecast.

The clustering notion for time-series data can be based on the divide and conquer method that has proven beneficial in engineering applications where large or complicated systems are solved by dividing large problems into smaller bits that are easy to solve [27].

The divide and conquer notion are employed in cluster learning, where a large dataset is clustered into distinct profiles, with each profile organised into a specific sequence for improved learning and forecasting; thus, the focus of this paper.

To this end, most of the clustering methods, as discussed in [19], [20], [21], and [22], fall under two categories:

1. Ensemble methods: Electric load is clustered into various patterns; each cluster pattern is passed to a learning algorithm for cluster forecasts. The cluster forecasts are then recombined into a single future forecast using an ensemble method. These have a similar drawback as most ensemble methods discussed above.
2. Feature extraction: These methods utilise clustering to extract features from the electric load data, in conjunction with times series data, which are passed to a learning algorithm for future forecasts.

The uniqueness of our proposed method is to provide individual cluster forecasts: this aims at clustering electric load into various consumption patterns. The most suitable cluster is selected for training to forecast future loads based on prior information of the coming day. Since there can be many clusters, deciding the right cluster to train for future forecasts is non-trivial. Also, after clustering, the long-term time-series relationship in the data is broken, making learning difficult. We propose a framework to reform the time series data after cluster learning to resolve the time-series dependency for cluster learning. There is also the need to model irregular time information of power consumption, non-linear patterns, and spatial information of all clusters for accurate load forecasting.

This paper proposes a novel sequence-to-sequence cluster learning and forecasting framework with the following contributions.

1. A new framework: Owing to the multimodal nature of electric load, K-means clustering with dynamic time wrapping (DTW) is used to identify different consumption patterns in the dataset; the proposed framework is designed to reform time series dependency after clustering. A CNN-GRU algorithm is used to adaptively learn spatiotemporal characteristics of each time-series cluster to improve load forecast accuracy.
2. Efficient inference: To identify the best cluster model to forecast future loads, an efficient LightGBM Classifier based on prior knowledge extracted from the future day.
3. Verification via real-world data: With a real-world historical dataset by Korea Power Exchange (KPX), we compare classical forecast methods and up-to-date learning algorithms by performing a case study on the dataset.

The rest of the contents are arranged as follows: Section II discusses the proposed framework. Section III describes the simulation process via a case study. Section IV presents analytical results and discusses them, and Section V summarises the results and conclusions.

II. PROPOSED FRAMEWORK

The proposed framework isolates load consumption patterns using clustering and reforms the time series dependency to produce a cluster sequence-to-sequence dataset, as depicted in Figure 1. This enables learning algorithms to focus more on each clustered pattern to improve forecast accuracy. This session discusses details of the proposed framework.

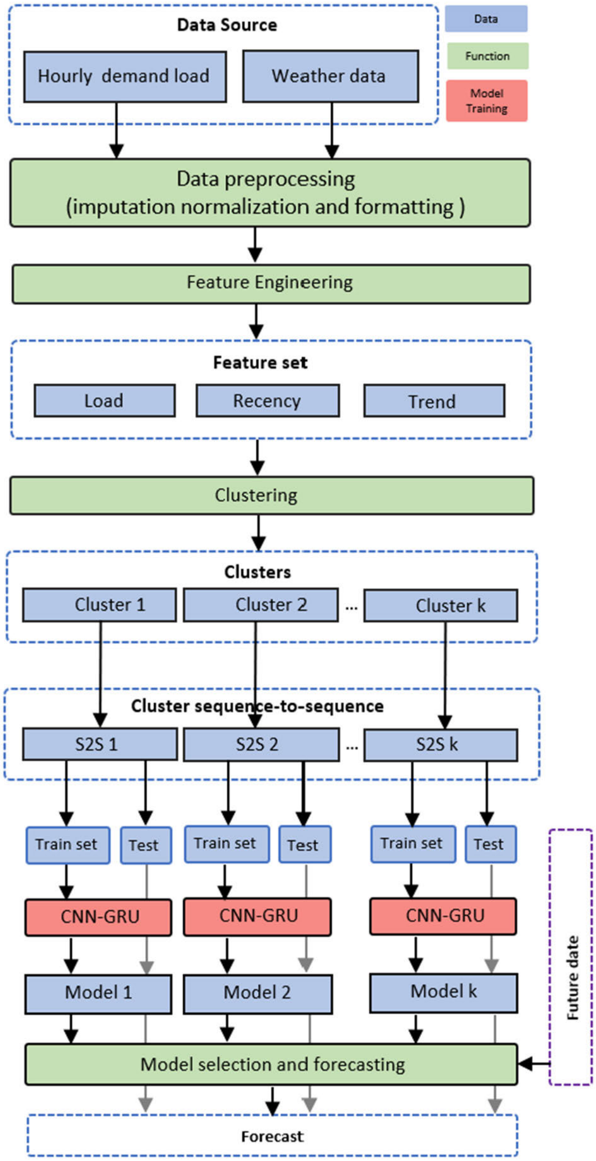


FIGURE 1. Proposed architecture.

A. DATA PRE-PROCESSING

In the current age of big data, processing high volume and diverse data is required since data suffer from anomalies such as missing data and outliers. When these are significant, they pose a problem for model learning; the data needs to be processed in advance to favour practical model training and forecasting. Times series noise in the form of missing data can be observed as an instance(time) record with ‘‘nan‘‘ or a blank value or a missing(absent) time record. These noise effects significantly affect forecast quality when the amount is significant. In this research, we resort to Copy-Paste

Imputation (CPI) technique which accounts for the total energy gaps in time series data using past data estimates [28].

Data values that might differ significantly from similar day/time expected values are outliers. In order to address the problem, there is a need to identify the outliers and find the root cause, if possible. In electric demand, outliers may occur due to equipment faults and catastrophes. These may be identified as a one-off or recurring event [29].

One effective method of identifying outliers is the generalised extreme studentised deviation (ESD) [30]. Generally, if the outlier is one-off and sporadic due to an error, we treat it as missing data and use the method discussed in [28] to resolve it. However, if the outlier is recurring, it is helpful to isolate and forecast them separately. Outliers, in this case, can be clustered, and their results are factored in as features of the dataset so that the forecasting algorithm can learn from the information [10]. This is where the proposed method shines, as clustered outlier information can be used as features; outliers can also be isolated and forecasted separately.

B. FEATURE ENGINEERING

From empirical evidence, factors that affect consumption can be categorised into weather, trend, and time series [28]. With the knowledge of these factors, future forecasts can be made even for multiple horizons ahead with less uncertainty. Feature engineering involves extracting useful information that our algorithm can easily interpret [10]. Table 1 details useful features considered in this paper.

- Weather factors

Weather factors influence electrical load consumption patterns significantly. Seasonal trends arise due to seasons such as winter and summer, where consumptions are highest due to the mass use of equipment like heaters and air-conditioners [31]. On the other hand, spring and fall seasons record low consumption as the weather is warmer, with less dependency on coolers or heaters.

- Trend factors

These factors influence load consumption due to a specific day, week, month, season, or event. These can be detailed as the differences in electricity consumption on working days identified as weekdays and non-working days as weekends and holidays [32]. Consumption differences can be narrowed down to a specific day with working and non-working hours. Commercial entities such as factory companies usually operate on working days, while shopping malls and other entertainment venues experience peak consumption on weekends and holidays; from a daily perspective, enterprise and entertainment venues experience higher consumption in the evenings than during the day. Residential units consume more energy in the mornings and evenings than in the afternoons, and midnight on working days, holidays and weekends have a complex consumption. As important as daily trends are, so are the transitions from one trend to another. Since notable trends can be identified for weekdays, weekends, and holidays, transitions, where a specific day is before or after a weekday, weekend, or holiday, are also vital information to consider.

TABLE 1. Feature set.

Factors	Feature	Description
Trend	Season	Sine/Cosine transform: T=365(6), f_t= day of a year
	Month	Sine/Cosine transform: T=12, f_t= month of a year
	Week	Sine/Cosine transform: T=52, f_t= week of a year
	Days in month	Sine/Cosine transform: depending on the month T={31,30,29,28}, f_t= day of a month
	Day of week	Sine/Cosine transform: T=7, f_t= day of a week
	Weekday/end	One-hot encode: weekday=1, weekend=-1
	Holiday	One-hot encode: holiday or not= {1,-1}
	After weekday	One-hot encode: day after weekday or not= {1,-1}
	Before weekday	One-hot encode: day before weekday or not= {1,-1}
	After holiday	One-hot encode: day after holiday or not= {1,-1}
	Before holiday	One-hot encode: day before holiday or not= {1,-1}
	Hour day	Sine/Cosine transform: T=24, f_t= hour of a day
	Working hour	One-hot encode: Working hours on a weekday or not={1,-1}
Recency	Lag 1~Lag 24	Normalised: Immediate past 24 hours load values, $t-lag$, $lag=\{1,2,...24\}$
Weather	Humidity	Normalised: Hourly atmospheric dryness
	Temperature	Normalised: Hourly temperature

- Recency factors

The electric consumption exhibits recency characteristics, thus indicating a high correlation between the immediate past and a scale factor [33]. In this paper, lag features are engineered to represent the correlation between electric consumption at a time instance and that of immediate past times. 24-hour lag horizons (lag 1~lag 24) are selected as features.

- Periodic factors

These factors reflect the cyclic nature of calendar days and times. Periodic features are extracted from time series load as a cyclic event [34]. Periodicity in electric consumption over the calendar days and trend factors such as season, month, year, and hour of a day is considered to represent the corresponding sine and cosine transformations:

$$c_1\,(t) = \sin(\tfrac{2\pi f_t}{T}) \tag{1}$$

$$c_2\,(t) = \cos(\tfrac{2\pi f_t}{T}) \tag{2}$$

where t is the time a feature value was observed; f_t is the value of a periodic feature at time t; T represents the period, e.g. for cycles observed in a year, $T = 365(6)$, weekly cycles $T = 52$, and daily cycle $T = 24$.

For the discrete features, one-hot encoding is used to encode feature values as 1 or -1. The summary of the features considered for training is shown in Table 1. Features are usually normalised before being fed to the neural network.

It is beneficial for the training process. It eliminates the difference in magnitude between the features in the dataset, improving network stability. Features that are not cyclic nor one-hot encoding are normalised between [1, −1] using the min-max normalisation (3).

$$x_n^* = \frac{x_n - x_{min}}{x_n - x_{max}}, \quad n = 1, 2, \ldots, N \tag{3}$$

where x_n^* is normalised dataset value;x_n is the n-th dataset sample; x_{min} and x_{max} represent the minimum and maximum of the dataset.

Considering the features discussed above, the load forecasts at a time t can be expressed as (4):

$$\hat{y}_t = \mathrm{f}\left(T_t^N\right) + \mathrm{f}\left(W_t^R\right) + \sum_{lag} \mathrm{f}(Y_{t-lag}) \tag{4}$$

where $\hat{y}_t$ is the future load forecast, $\mathrm{f}\left(T_t^N\right)$ is a function of trend factors, N is the number of trend factors, $\mathrm{f}\left(W_t^R\right)$ is a function of weather factors, R is the number of weather factors, $\mathrm{f}(Y_{t-lag})$ is a function of recency factors and Y_{t-lag} is the consumption at $t - lag, and\ lag = \{1, 2 \ldots 24\}$.

C. FEATURE EXTRACTION

According to the features summarised in Table 1, we determine and select features that strongly correlate to the electric load. This is necessary because although many features can improve forecast accuracy, they can also adversely affect the training time and the variance of the model.

The correlation between dataset features and load consumption patterns is evaluated via Distance correlation (DC). DC measures the association between paired random variables of linear or non-linear nature with flexible dimensions. The DC coefficient of a paired variable is zero if and only if they are independent [35]. The confusion correlation matrix between the electric load and the d-th feature is evaluated using (5). This is the ratio of relative distance covariance between the paired variables to the product of their standard deviation.

$$\mathrm{dCor}\left(\boldsymbol{X}^d, \boldsymbol{Y}\right) = \frac{\mathrm{dCov}^2(\boldsymbol{X}^d, \boldsymbol{Y})}{\sqrt{\mathrm{dVar}(\boldsymbol{X}^d)\mathrm{dVar}(\boldsymbol{Y})}} \tag{5}$$

$$\mathrm{dCov}^2\left(\boldsymbol{X}^d, \boldsymbol{Y}\right) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} A_{i,j}B_{i,j} \tag{6}$$

$$\mathrm{dVar}^2\left(\boldsymbol{X}^d\right) = \frac{1}{n^2}\sum_{i=1}^{n} A_{i,j}^2 \tag{7}$$

$$\mathrm{dVar}^2\left(\boldsymbol{Y}\right) = \frac{1}{n^2}\sum_{i=1}^{n} B_{i,j}^2 \tag{8}$$

$$A_{i,j} = a_{i,j} - \bar{a}_i - \bar{a}_j - a_{..} \tag{9}$$

$$B_{i,j} = b_{i,j} - \bar{b}_i - \bar{b}_j - b_{..} \tag{10}$$

$$a_{i,j} = \left\| X_i^d - X_j^d \right\|, \quad i, j = 1, 2, \ldots, n \tag{11}$$

$$b_{i,j} = \left\| Y_i - Y_j \right\|, \quad i, j = 1, 2, \ldots, n \tag{12}$$

where $\boldsymbol{X}^d$ is the d-th feature, $\boldsymbol{Y}$ is the target label, (X_i^d, Y_i) are the ith sample from the paired random variable (X^d, Y), $i = 1, 2, \ldots, n$; $\bar{a}_i$ is mean of the i-th row; $\bar{a}_j$ is mean of the j-th column; $a_{..}$ and $b_{..}$ are the grand mean of the distance matrix of samples $\boldsymbol{X}^d$ and $\boldsymbol{Y}$ respectively; $a_{i,j}$ and $b_{i,j}$ are pairwise distance matrices of sample $\boldsymbol{X}^d$ and $\boldsymbol{Y}$ respectively. The range of value for $dCor(\boldsymbol{X}^d, \boldsymbol{Y})$ is [0,1]. When $dCor(\boldsymbol{X}^d, \boldsymbol{Y})$ equals 0, this signifies the d-th feature and the label do not correlate; when $dCor(\boldsymbol{X}^d, \boldsymbol{Y})$ equals 1, it signifies a strong correlation between the d-th feature and the label. From (13), V^d is an identification function for the d-th feature. If the correlation coefficient is greater than or equal to a set value ω, V^d is 1, and the d-th feature value is included as part of the final input dataset; if the correlation coefficient is less than ω, V^d is 0, and the d-th feature is discarded.

$$\mathrm{V}^d = \begin{cases} dCor\left(\boldsymbol{X}^d, \boldsymbol{Y}\right) & \geq \omega \mathrm{V}^d = 1 \\ dCor\left(\boldsymbol{X}^d, \boldsymbol{Y}\right) & < \omega \mathrm{V}^d = 0 \end{cases} \tag{13}$$

D. K-MEANS CLUSTERING

To isolate load consumption patterns from the given dataset, K-means clustering is employed. Clustering is a popular machine-learning technique to identify patterns in a dataset [36]. K-Means is greatly popular due to its simplicity, interpretability, and fast convergence [37].

With a dataset $\boldsymbol{X} = [x_1, x_2, \ldots, x_n] \in \boldsymbol{R}^{p \times n}$ the objective is to partition the dataset into K clusters μ_k, where $\mu_k \in \boldsymbol{R}^p$ is the model associated with the k^{th} cluster, and $k = 1, 2, \ldots K$. A set of identification functions $c_{ik} \in \{0, 1\}$ represent assignments where $c_i = [c_{i1}, c_{i2}, \ldots, c_{ik}]^t$ is the k^{th} canonical basis vector in $\boldsymbol{R}^K$ if and only if x_i belongs to the k^{th} cluster. Here $\mu = \{\mu_k\}_{k=1}^K$ is the cluster centres and $c = \{c_i\}_{i=1}^n$ are associated data samples. The objective of K-means clustering is to minimise the sum of the squared Euclidean distances of samples belonging to a clusters [38]:

$$\mathrm{J}\left(c, \mu\right) = \sum_{i=1}^{n}\sum_{k=1}^{K} \mathrm{c_{ik}} \left\| \mathrm{x_i} - \mu_\mathrm{i} \right\|_2^2 \tag{14}$$

where $J(c, \mu)$ is the objective function.

To identify samples belonging to a cluster, the centroids and the distance between two points must be calculated.

Given that (15) and (16) represent time-series instances:

$$\boldsymbol{X} = [x_1, x_2, \ldots, x_n] \tag{15}$$

$$\boldsymbol{Y} = [y_1, y_2, \ldots, y_n] \tag{16}$$

where n is the number of samples in the time series; the Euclidean distance between $XandY$ is obtained via:

$$\mathrm{d_{euc}}\left(\boldsymbol{X}, \boldsymbol{Y}\right) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_n - y_n)^2} \tag{17}$$

Euclidean distance is not suited for time series data as the latter has a high dimensional form which may lead to information loss [10]. Dynamic time wrapping (DTW) is much suited for time-series analysis as it flexibly maps two sets of time series to obtain their relative distance [39].

The distance between the two time-series $\mathrm{d_{dtw}}\left(\boldsymbol{X}, \boldsymbol{Y}\right) = \mathrm{d_{dtw}}\left(x_n, y_n\right)$ is evaluated as the cost of the optimal alignment path:

$$\mathrm{d_{dtw}}\,(x_i, y_i) = \mathrm{d_{euc}}\,(x_i, y_i) + \min[d_{dtw}\,(x_{i-1}, y_{i-1})\,, \times \mathrm{d_{dtw}}\,(x_i, y_{i-1})\,, \mathrm{d_{dtw}}\,(x_{i-1}, y_{i-1})] \tag{18}$$

To maintain the shape characteristics, the centroids of DTW are obtained via dynamic time-warping barycenter averaging (DBA) [40].

This paper employs K-means clustering with DTW to cluster historical daily load profiles into K clusters. Each cluster contains similar demand characteristics and patterns. A member of a cluster is denoted X_i^k, $k = 1, \ldots K$, $i = 1, \ldots r, r \in \{n_1, n_2, \ldots, n_K\}$, n_k is the total number of daily profiles that belong to the $k - th$ cluster, as depicted in Figure 2(a). From Figure 2(a), each daily profile A, B, C, D, E, and F is of the form X_i^k which is a vector in $\mathbb{R}^{24}$, this represents a day load and is characterised by a 24-hour sequence, $X_i^k = \left[X_{i,1}^k, X_{i,2}^k, \ldots, X_{i,24}^k\right]$. The position of X_i^k in the original time series is determined by the date of observation d_t.

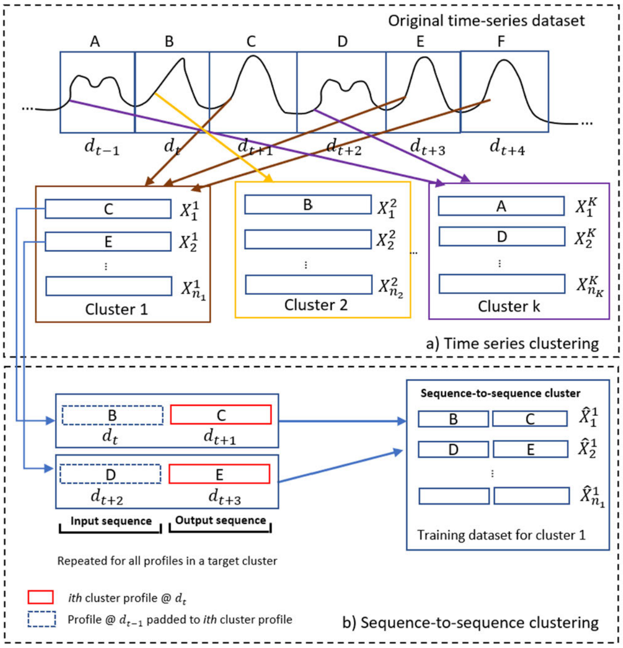


FIGURE 2. Sequence-to-sequence (S2S) cluster formulation.

E. SEQUENCE-TO-SEQUENCE (S2S) CLUSTER MODELING

After the dataset has been clustered into groups of similar load consumption patterns, the clusters cannot be used directly to learn a model for future forecasts since the long-term time-series dependency in the dataset is lost, and forecasting becomes impractical. Time series measures the sequence of values over time where values at any point in the set are strongly correlated to previous values. Given a data x_t at time t, the objective of time-series forecasting is to anticipate the upcoming value x_{t+1} at time $t + 1$ or x_{t+n} at time $t + n$, which models the history of the dataset considering factors that affect the dataset. From a sequence-to-sequence (S2S) perspective, given a past data sequence of horizon h, $\boldsymbol{X} = [x_{t-h}, \ldots, x_{t-2}, x_{t-1}]$, the forecast objective is to estimate a future data sequence of horizon p, $\boldsymbol{Y} = [y_t, y_{t+1}, y_{t+2}, \ldots, y_{t+p}]$, where h and p are desired horizons for past and future datasets.

To enforce cluster learning, S2S cluster formatting is explored to transform the dataset of each cluster k into a training dataset. To achieve this, the clustered dataset is transformed into a supervised learning format, $\boldsymbol{X}$, $\boldsymbol{Y}$, where $\boldsymbol{X}$ is the sequence of load data set as the input and $\boldsymbol{Y}$ is the corresponding sequence set as the label or output to $\boldsymbol{X}$. Each X_i^k member of a cluster observed at a time instance d_t is set as the target output $\boldsymbol{Y}$ as it is the sequence we want to predict given the input $\boldsymbol{X}$. The input sequence is obtained as the profile that precedes X_i^k in the original time series. If X_i^k was observed on a day d_t, then the profile that precedes it was observed on $dt - 1$.

The profile at d_{t-1} is padded with that of d_t, thus restoring the time-series dependency, as shown in Figure 2(b). This process is repeated for all the profiles in a cluster. The resulting dataset is the sequence-to-sequence cluster dataset for training the machine learning algorithm on a specific cluster c^k.

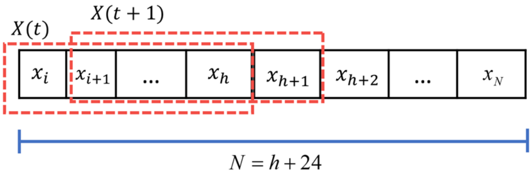


FIGURE 3. Sequence sliding window with length h.

The sliding window method is employed to section the S2S cluttered dataset into subsets with the matching length, which is fed as the input sequence to the CNN-GRU model. With a sliding window of length h, suppose $\boldsymbol{X}_t = [x_i, x_{i+1}, \ldots, x_h]$ is an input set at time t with length h, then the input at the next time step $t + 1$ will be $\boldsymbol{X}_{t+1} = [x_{i+1}, x_{i+2}, \ldots, x_{h+1}]$, where $t \geq 1$ and $h > 1$, as shown in Figure 3. The final input sequences are obtained from an Nth sized dataset, divided into N-h+1.

F. CNN-GRU MODEL

In this paper, the learning algorithm of choice is the hybrid CNN-GRU algorithm employed to learn spatiotemporal features in each S2S cluster dataset.

- CNN

These are special neural networks developed for processing spatial or time-series data, such as energy consumption data, considered as a 1D grid [34].

The 1D integrated CNN network consists of input, pooling, padding, convolution, and a linear layer. One objective of CNN is identifying trends from adjacent values in a time series dataset via the convolution operation [41]; this is expressed as in (19):

$$FM_i = (I * K)_i = \sum_m I_{i+m} \cdot K_m \tag{19}$$

where "*" is the convolution operation. K is a 1D kernel, I is the 1D input of a target layer, m is the kernel size and FM_i also referred to as a feature map, is the output of the convolution operation. The feature map is via the convolution operation using a persistent kernel on input sequentially; thus, a feature can be sensed and learned, irrespective of its location within the input sequence. Applying the convolution operation (19) on input data $x_1, x_2, \ldots, x_n$ transforms it into feature maps $FM_1, FM_2 \ldots, FM_n$. Before the convolutional operation, pooling and a padding operation are applied to the input, respectively. The pooling layer is employed to decrease the spatial size of the input, making the feature dimension smaller and thereby reducing the number of parameters in the network. This helps combat overfitting. Pooling makes the network robust to input distortions as it aggregates (max, sum, average) the neighbourhood values of the input. The aggregating function employed in this paper is max (aka MaxPooling), a down-sampling scheme. To apply MaxPooling, a sliding filter transitions across the input sequence, where the maximum value of the overlapped area is set as the output. Padding is adding one or more zeros around the input boundaries to increase its effective size. Padding helps retain more information by preserving the size of the input.

- GRU

RNN generalises feedforward networks as sequences [42]. For an input $\boldsymbol{X} = [x_1, x_2, \ldots, x_n]$, RNN evaluates an outputs sequence $\boldsymbol{Y} = [y_1, y_2, \ldots, y_n]$ via iteration using the following equations:

$$h_t = \text{sigm}(W^{hx} x_t + W^{hh} h_{t-1}) \tag{20}$$

$$y_t = W^{yh} h_t \tag{21}$$

RNNs have incredible success handling S2S tasks [43], where a decision at a time step $t-1$ is affected by that of a time step t, signifying a temporal dependency. However, RNNs suffer from vanishing and exploding gradients and cannot capture long-term dependencies effectively [44], [45].

Gradient vanishing refers to the case where the gradient norm for long-term relationships decreases exponentially to zero, inhibiting the learning of long-term temporal relationships. In contrast, gradient exploding is the contrary event. An approach to mitigate these problems is using a more sophisticated activation function employed by the LSTM unit [46]. A variant of LSTM dubbed GRU can capture long-term dependencies. It is immune to the vanishing gradient needing fewer calculations to update its hidden state. Figure 4. shows a GRU cell unit. The update gate controls the amount of memory retained by the network, and the reset gate coordinates the input and memory data.

The GRU gating mechanism can be formulated as:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{22}$$

$$r_t = \sigma(W x_t + U_r h_{t-1} + b_r) \tag{23}$$

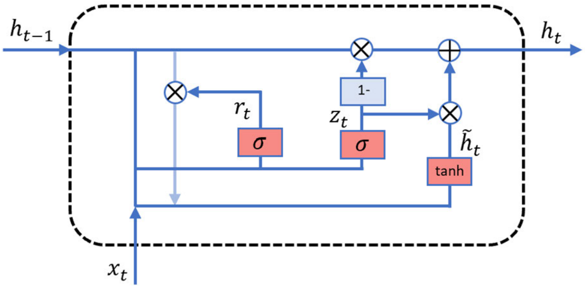


FIGURE 4. Structure of GRU Cell.

$$\tilde{h}_t = tanh(W x_t + U(r_r \odot h_{t-1} + b_z) \tag{24}$$

$$ht = (1 - z_t)\, h_{t-1} + z_t \tilde{h}_t \tag{25}$$

where z_t is the update gate, r_t is the reset gate, $x_t \in \mathbb{R}^N$ represents the input sequence at time t. $h_t, h_{t-1} \in \mathbb{R}^H$ are the current and previous hidden state, respectively. In (22)-(24), the notation σ is a sigmoid function, $[W_z, U_z, b_z]$, $[W_r, U_r, b_h]$, and $[W, U, b_h]$ are parameters of the update, reset, and hidden modulation gates responsible for learning. In (24), the operator $\odot$ does an element-wise multiplication.

- CNN-GRU Network Configuration

The proposed network configuration for CNN-GRU comprises three lines that capture different non-linear features from the dataset using different CNN configurations Figure 5. The first line of the network passes the clustered sequence dataset to a padding layer; after 1D convolution, padding is applied to preserve the input dimension; this helps maintain the information at the input boundaries. The 1D convolutional layer is connected to a second padding layer flowed by a second 1D convolution; the resulting feature map is passed to a GRU network. The second line applies the pooling operation to the clustered sequence dataset; this layer entails sliding the kernel over the input to return the average value. The pooling operation samples the feature map, reducing network complexity. Padding is subsequently applied to the feature map and fed to the 1D convolutional layer. The padding and coevolution operations are repeated, and the resulting feature map is passed to a GRU network. The third line repeats the second line but with a relatively different filter and kernel size. The three lines extract spatial representation in the dataset, passing on to the three GRU.

After extracting the spatial features, the cluster feature maps are fed into GRU layers. The GRU layers model temporal characteristics; the outputs are concatenated into a linear layer to forecast future load consumption.

This process is repeated for each clustered sequence dataset to obtain a forecast model for each cluster.

G. FORECAST MODEL SELECTION

Forecasting a future date is difficult because the cluster that this day will fall under is unknown beforehand. To determine the cluster of a future date, a classification algorithm LightGBM is employed. Classification models a predictive problem where a label is predicted by input data [47]. The

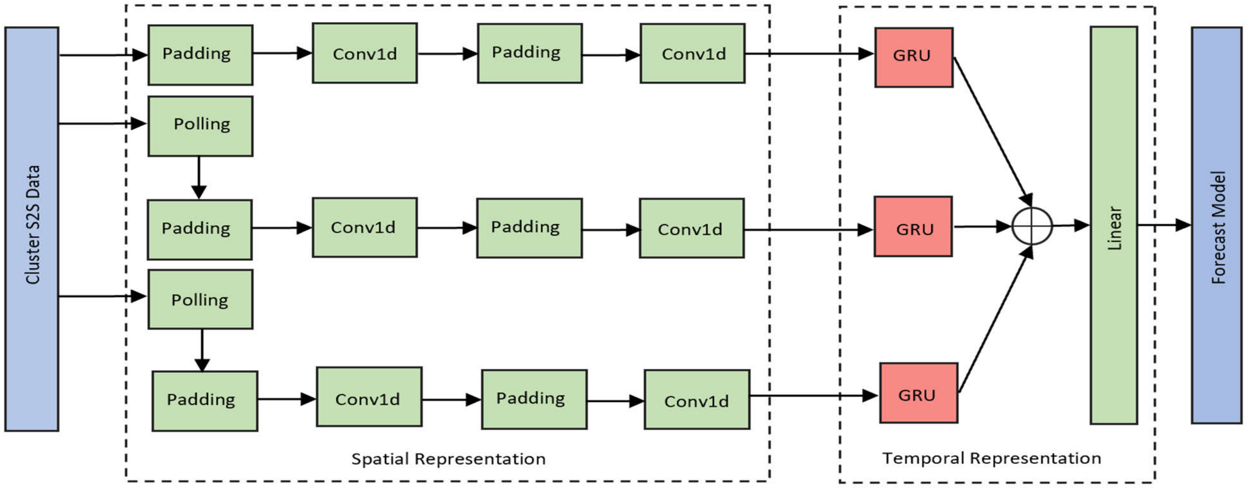


FIGURE 5. CNN-GRU network architecture.

classification algorithm takes in a multivariate dataset and a label as a requirement. The clustered dataset developed in section II-B is used to model a classifier in our proposed methodology. Based on K-means clustering, each record in the dataset is assigned a membership C^k that is the record's label. LightGBM, which is a gradient-boosting decision tree (GBDT) method [48], is employed for future load classification due to lower memory usage, higher efficiency, and faster training time [49].

III. CASE STUDY

The case study experiments were conducted using the Jupyter Lab with Python3 on a PC with an Intel i-9 processor of 5.0 GHz and 32 GBs of RAM.

A. DATA DESCRIPTION

The proposed study was verified for efficacy using a nationwide dataset via KPX [50]. The electric load data is a multivariate time series spanning 2015 to 2022 with an hourly resolution. An additional weather dataset spanning 2015 to 2022 of an hourly resolution was obtained to feature the electric load dataset via an open API provided by Korea public data portal [51]. Compared to conventional methods, the data is processed for missing data and feature engineering based on events and characteristics that affect electric load consumption, as detailed in section II-B. Distance correlation is used to identify features that strongly correlate to electric load consumption, thereby reducing the dimension of the feature dataset: this helps refine the feature dataset and contributes to a reduction in training time. Figures 6 to 8 show the correlation results for trend and demand, cyclic and demand, and recency factors and demand, respectively. Figures 6 and 7 depict the distance correlation between trends, cyclic factors, and demand. Here DC shows a low correlation between the factors and demand, with a maximum of about 0.4. This is because DC is biased towards short-term (hourly) correlation, as such factors with long-term correlation (hour, day, week, month, seasonal) though significant, are assigned low correlation indices. To resolve this, we grouped factors with similar characteristics; for example, hour, day, week, month, and seasonal features are grouped independently. For each group, the probability of each factor is evaluated based on their distance correlation values, and a discrimination threshold of 0.5 is set to decide whether the feature is significant.

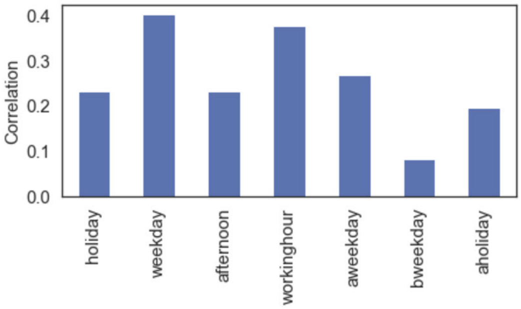


FIGURE 6. Correlation between trend factors and demand.

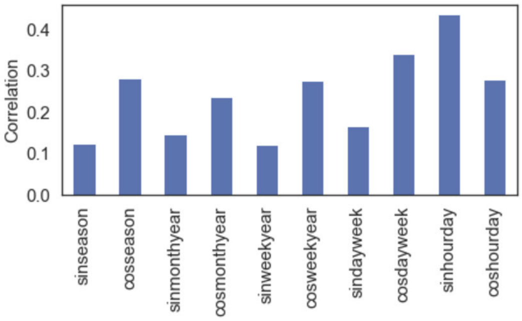


FIGURE 7. Correlation between cyclic factors and demand.

Figure 8 depicts the DC between recency factors and demand. Because load consumption is cyclic for a day, events close to the start and the end of the cycle are

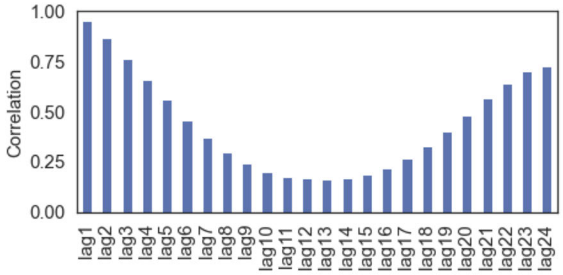


FIGURE 8. Correlation between recency factors and demand.

similar; consequently, Figure 8 shows a parabolic trend. From the figure, demand values close to each other are strongly correlated, but the correlation decreases stepwise feather away. Demand values 11 to 14 hours apart exhibit the least correlation, but the correlation increases steadily at the end of the cycle. Using a discrimination threshold of 0.5, lag 1 to 6 and lag 20 to 24 are selected as significant recency features. Temperature and humidity had a 0.8 and 0.45 correlation index with load for weather factors; as such, the temperature feature is retained. With the dataset processed into a multi-variate feature dataset, the dataset is partitioned into 90% training dataset and 10% test dataset. K-means clustering with dynamic time wrapping is employed to extract non-linear load consumption patterns over a day horizon from the training dataset. The average silhouette [52], elbow curve [53], and gap statistic [54] are used to estimate the best number of clusters [55].

The average silhouette and elbow curve methods are termed direct methods, and the gap statistic is an example of a statistical method. The direct methods minimise within-cluster error, while statistical methods collect evidence to support or reject a null hypothesis [56].

We employ a combination of these indices to select the number of clusters with the most occurrence among all indices. Figure 9 depicts the elbow method. A sharp decline in the sum of squared distances is observed at $K = 3$ therefore, 3 is selected as the optimum number of clusters.

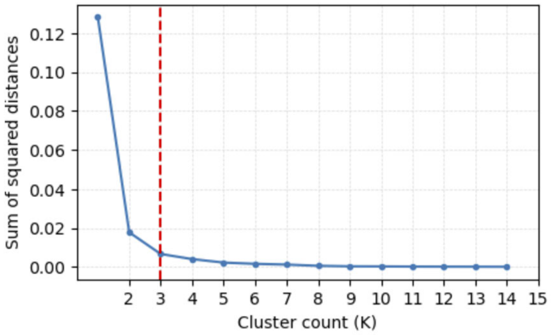


FIGURE 9. Elbow curve.

Figure 10 shows the three normalised daily cluster patterns obtained from K-means. The clusters also give insight into consumption patterns for each day in a week and each month in a year, as in Figure 11 and Figure 12, respectively. From Figure 11, it is clear that cluster 3 is very dominant during the weekday (Monday-Friday) and less prevalent on Saturday; cluster 3 patterns are not recorded on Sundays. Clusters 2 and 1 are weekend clusters more dominant on weekends. Figure 13 shows that cluster 3 is dominant across 12 months; however, clusters 1 and 2 show an interesting pattern. Cluster 2 is dominant from January to May, with a dip from June to September and a rise from October to December. This signifies a cluster for the cold season. On the flip side,

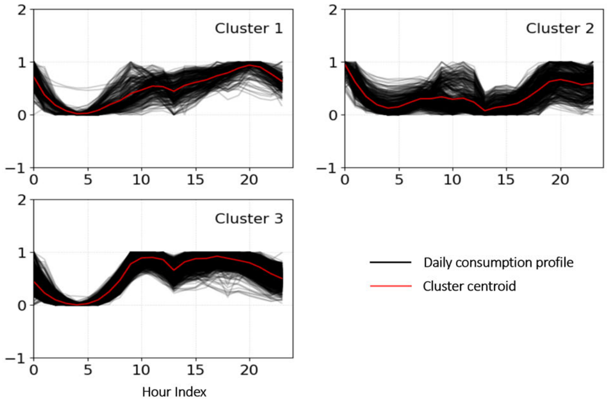


FIGURE 10. Normalised daily cluster patterns with centroids.

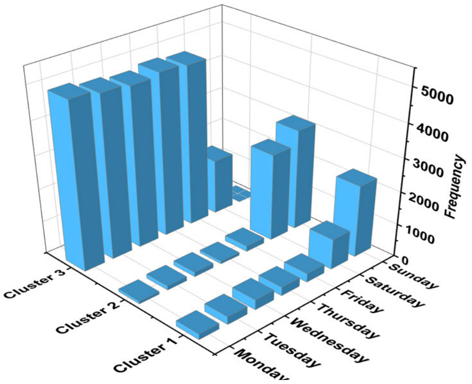


FIGURE 11. Daily cluster distribution.

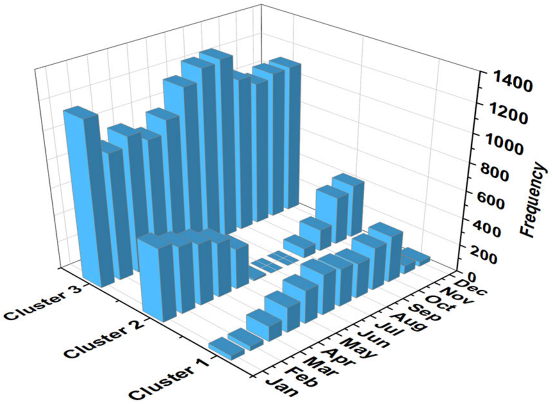
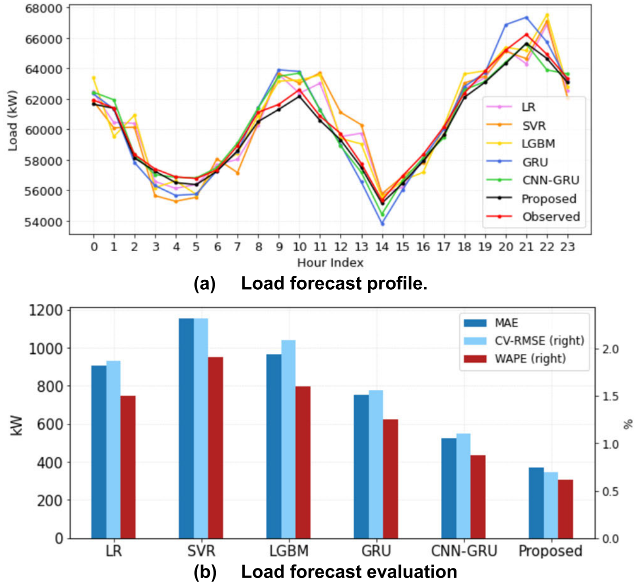


FIGURE 12. Monthly cluster distribution.



(a) Load forecast profile.

(b) Load forecast evaluation

FIGURE 13. Analysis of load forecast for a day in cluster 1.

TABLE 2. The network configuration of CNN-GRU.

		Layers	Activation shape
	Input	Input	(None,48, 42)
Line 1	CNN	Padding	(None,48, 54)
		Conv1d	(None,128, 54)
		Padding	(None,128, 54)
		Conv1d	(None, 128, 48)
	GRU	Input	(None,48, 128)
		GRU	(None, 128)
Line 2	CNN	Average Polling	(None, 42, 24)
		Padding	(None,42, 28)
		Conv1d	(None,128, 24)
		Padding	(None, 128, 24)
		Conv1	(None, 128, 24)
	GRU	Input	(None,24, 128)
		GRU	(None,128)
Line 3	CNN	Average Polling	(None,42, 12)
		Padding	(None,42, 14)
		Conv1d	(None,128, 12)
		Padding	(None,128, 14)
		Conv1d	(None,128, 12)
		Input	(None,128, 12)
	GRU	GRU	(None,128)
	Concatenate	Input	(None,128), (None,128), (None,128)
		Output	(None,384)
		Linear	(1)

cluster 1 kicks in slowly from March, with a high from May to October and then a low during November and December. Thus, complex consumption patterns can be identified for load forecasting with clustering. After clustering, each cluster dataset is processed into a sequence-to-sequence format to enable the learning algorithm to learn patterns in a particular cluster and forecast multiple steps. This gives the model a better shot at a higher accuracy.

B. SPATIOTEMPORAL FORECASTING

CNN-GRU is configured to learn spatiotemporal features in the S2S cluster dataset. Table 2 lists the network configuration for CNN-GRU. It comprises three lines, polling layer, padding layer and convolution layer for spatial representation. Each line is then fed to a GRU network to model temporal characteristics. Finally, the output of each GRU unit is concatenated and linearised. CNN-GRU is used to learn a model for each cluster dataset. With a model for each cluster, it is non-trivial to know which model forecasts a given day best; however, prior knowledge of the future date trend factors, as described in section II-B, can be obtained as predictors for identification.

LightGBM is used to learn a classification model using the predictors from each dataset with cluster membership as a label. For the LightGBM classification model, we used 29 features from the trend factor, as discussed in section II-B. With a dataset set size of 2606 daily profiles spanning eight years, the LightGBM classification model produced an accuracy of 95% when identifying future date cluster membership.

C. EVALUATION METRICS

This paper uses LR, SVR, GRU, and CNN-GRU models on the dataset for comparative analysis. To evaluate the forecast methods, we determine how well the forecast matches the observed load. To achieve this, we evaluate the deviation between the observed load sequence and the corresponding forecast sequence via Mean Absolute Error (MAE), Coefficient of Variation of the Root Mean Squared Error (CV-RMSE), and Weighted Average Percentage Error (WAPE) [57]. Compared to Mean Absolute Percentage Error (MAPE) [58], WAPE weighs the individual absolute errors to account for the intermittent load consumption; as such, WAPE is recommended over MAPE. MAE, CV-RMSE, and WAPE are evaluated using (26) - (28).

$$MAE = \frac{1}{N} \sum\nolimits_{i=1}^{N} \left| y_i - \hat{y}_i \right| \tag{26}$$

$$CV - RMSE = \sqrt{\frac{\sum_{i=1}^{N} \left(y_i - \hat{y}_i \right)^2}{N}} / \frac{\sum_{i=1}^{N} y_i}{N} \tag{27}$$

$$WAPE = \frac{\sum_{i=1}^{N} \left| y_i - \hat{y}_i \right|}{\sum_{i=1}^{N} \left| y_i \right|} \tag{28}$$

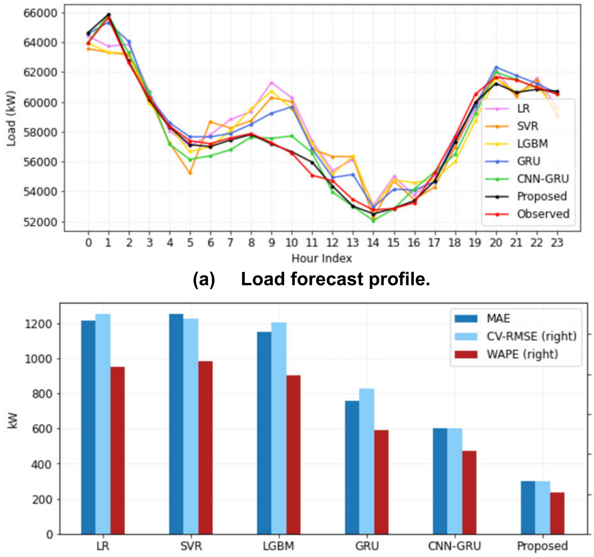
where y_i and $\hat{y}_i$ are the observed and forecasted load values at the time step i, and N is the number of samples in a sequence spanning a time horizon.

IV. RESULTS AND DISCUSSIONS

In this section, the forecast results obtained by the proposed method is compared with benchmark algorithms such as LR [5] and SVR [6] and also with state-of-the-art forecast provided by LGBM, GRU and CNN-GRU, [10], [14] and [20] respectively. Figures 13 – 19 present a comparative result of the forecast algorithm and how they fare with the proposed method.

A. CLUSTER 1 FORECAST RESULTS

Figure 13 presents a forecast analysis of a consumption pattern under cluster 1. Cluster 1 contains mainly weekend consumption and also consumption between April and October. From the figure, the forecast by the proposed method (CNN-GRU with clustering) tracks the observed values best compared to CNN-GRU, GRU, LGBM, SVR, and LR. The forecast evaluations show that SVR had the worst performance, followed by LR, LGBM, GRU, and CNN-GRU in succession with WAPE of 1.91%, 1.50%, 1.59%,1.25%, and 0.87%, respectively, the proposed method with a WAPE of 0.61%.



(a) Load forecast profile.

(b) Load forecast evaluation

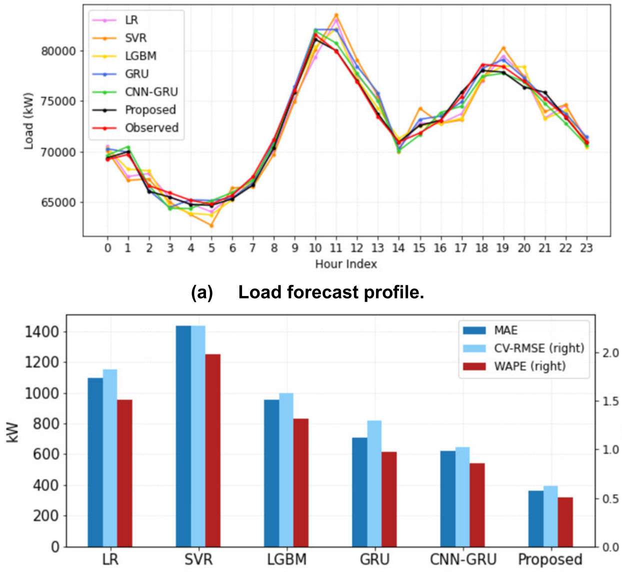
FIGURE 14. Analysis of load forecast for a day in cluster 2.

B. CLUSTER 2 FORECAST RESULTS

Figure 14 presents a forecast of consumption patterns that fall under cluster 2. Profiles under cluster 2 are mostly weekend consumption between January and May and September to December. The figure shows that forecasts from the proposed study fit the observed values best. SVR had the worst performance, followed by LR, LGBM, GRU, and CNN-GRU in succession with WAPE of 2.15%, 2.09%, 1.99%, 1.30%, and 1.00%, respectively, and the proposed method with a WAPE of 0.51%.

C. CLUSTER 3 FORECAST RESULTS

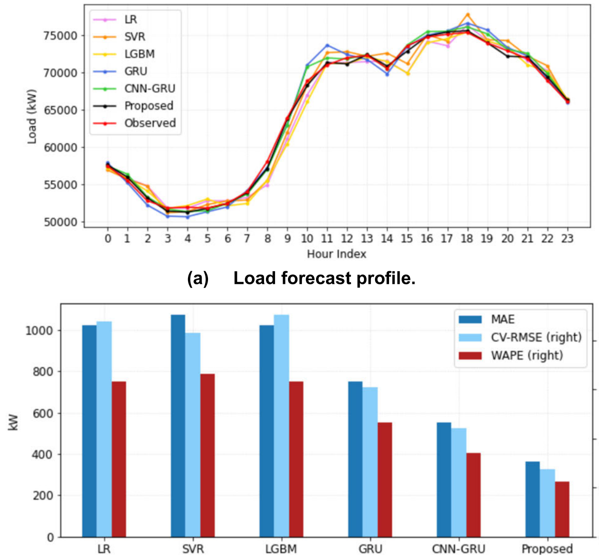
Figure 15 depicts the forecast results of a consumption profile tahe fall under cluster 3; these are primarily available throughout the year as weekday consumptions with a low presence on Sundays and absence on Sundays. Figure 15(b) shows a WAPE of 1.98%, 1.51%,1.32%, 1.00%, 0.86% and 0.50% for SVR, LR, LGBM, GRU, CNN-GRU and proposed, respectively.



(a) Load forecast profile.

(b) Load forecast evaluation

FIGURE 15. Analysis of load forecast for a day in cluster 3.



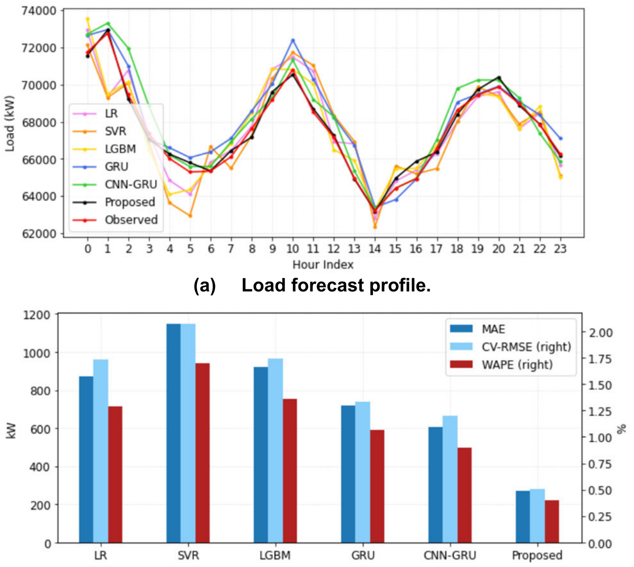
(a) Load forecast profile.

(b) Load forecast evaluation

FIGURE 16. Analysis of load forecast for a weekday.

D. WEEKDAY FORECAST RESULTS

Figure 16 illustrates the forecast profile of a typical weekday consumption. It shows that almost all the algorithms can fit the observed load closely; however, the proposed method scores the least WAPE of 0.56%, followed by CNN-GRU, GRU, LGBM, SVR and LR with WAPE of 0.85%, 1.16%, 1.58%, 1.58% and 1.66%, respectively.



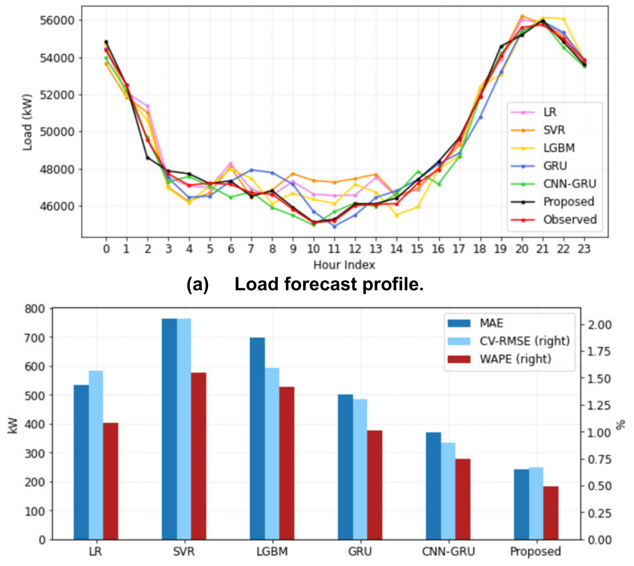
(a) Load forecast profile.

(b) Load forecast evaluation

FIGURE 17. Analysis of load forecast for a weekend.

E. WEEKEND FORECAST RESULTS

Figure 17 illustrates the forecast results on a typical weekend consumption profile. From the evaluation in Figure 17(b), the proposed method scored the best WAPE of 0.39% and CNN-GRU, GRU, LR, LGBM, and SVR scored WAPE of 0.90%, 1.06%, 1.28%, 1.36% and 1.70%, respectively.



(a) Load forecast profile.

(b) Load forecast evaluation

FIGURE 18. Analysis of load forecast for a holiday.

F. HOLIDAY FORECAST RESULTS

Figure 18 exemplifies a holiday consumption profile.

Here the proposed method produced a WAPE of 0.48%, CNN-GRU and GRU scored 0.75% and 1.01%, LR scored 1.08%, LGBM scored 1.42%, and SVR scored 1.55%.

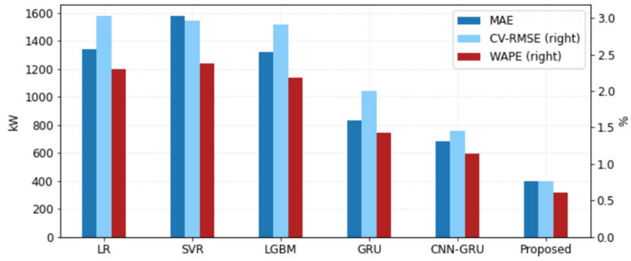


FIGURE 19. Analysis of load forecast for the test dataset.

G. FORECAST RESULTS ON TEST DATASET

Figure 19 demonstrates the results of training the various forecast algorithms on 90% training dataset and forecasting the remaining 10% test dataset. Generally, the proposed method produced a WAPE of 0.61%, CNN-GRU and GRU scored 1.14% and 1.43%, LGBM scored 2.18%, LR scored 2.30%, and SVR scored 2.37%. This shows that the proposed method records the best accuracy; thus, it effectively improves the accuracy of CNN-GRU on the test dataset by about 50%.

H. DISCUSSIONS

From the results, SVR mostly produced the least accuracy, followed by LR and LGBM, compared to GRU, CNN-GRU, and the proposed method. From the evaluation results, as shown in Figures 16 to 18, GRU shows that weekday forecasts have better accuracy than weekends and holidays. This is because weekday profiles are dominant with less variability, and weekends and holidays have lesser samples and exhibit more variability. Generally, forecasts around the turning points of the profile are more erroneous, as indicated by LR, SVR, and GRU.

Since CNN-GRU is an upgrade of GRU, the hybrid model can learn spatial-temporal features to improve GRU forecast around the turning points of the consumption curve.

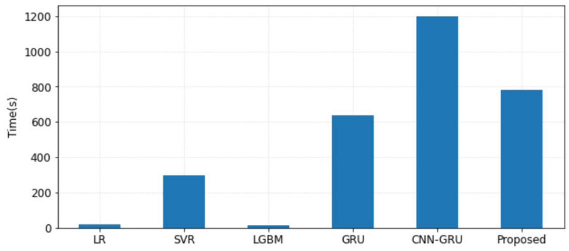


FIGURE 20. Comparative analysis of training time for the forecast methods.

The proposed method has the best WAPE, and the values are stable for weekdays, weekends and holidays. Although CNN-GRU can improve the accuracy of GRU, it incurs a higher training time due to the added spatial feature extraction layer. The proposed sequence-to-sequence cluster method can mitigate the shortcomings of both GRU and CNN-GRU by 1) Clustering the dataset into clusters to extract complex patterns for the learning algorithm to focus on; in doing so, the training time of the algorithm is reduced since only a subset of the dataset is used to train a cluster. 2) each S2S cluster is trained to learn spatiotemporal representation to improve forecast accuracy. From a model training time perspective, LGBM is the fastest, followed by LR. SVR was faster than GRU, and CNN-GRU followed with the worst time. Due to dataset size and the convolution operation, CNN-GRU exhibits the highest training time; but the proposed method can reduce this training time due to clustering by about 35%. Figure 20 illustrates the training time for the individual forecast algorithms.

V. CONCLUSION

One vital aspect of load forecasting is identifying features that characterise the data; this includes complex relationships, trends, cycles, and recency factors.

This paper proposes an S2S cluster learning algorithm for electric load forecasting designed to reform time series dependency after clustering. The dataset extracts spatial and complex features to improve load forecasting accuracy. A case study uses a historical Korea Power Exchange (KPX) dataset to train the proposed algorithm. Three (3) consumption patterns are identified using K-mean clustering with DTW from the dataset. Each cluster is formatted into an S2S dataset using the proposed method to restore time-series dependency, making it suitable for training. The forecast model obtained from training the S2S cluster dataset with CNN-GRU is evaluated for efficacy by comparing it to benchmark models such as LR, SVR, LGBM, GRU, and CNN- GRU. Based on the study, the proposed method yielded the highest forecast accuracy with a WAPE of 0.61% compared to that of LR, SVR, LGBM, GRU, and CNN-GRU with WAPE of 2.30%, 2.37%, 2.18%, 1.43%, and 1.14% respectively on the test dataset. The proposed method also speeds up the training process of the forecast algorithm by about 35%, given that only a subset of the dataset is trained due to clustering. These contributions are reached under a comprehensive dataset; however, the model might suffer from high bias under a scanty dataset. Future work will follow transfer learning [25] to adapt the model to a scanty dataset based on the results.

REFERENCES

[1] M. Santamouris and K. Vasilakopoulou, "Present and future energy consumption of buildings: Challenges and opportunities towards decarbonisation," *e-Prime*, vol. 1, Oct. 2021, Art. no. 100002.

[2] A. Zakari, F. F. Adedoyin, and F. V. Bekun, "The effect of energy consumption on the environment in the OECD countries: Economic policy uncertainty perspectives," *Environ. Sci. Pollut. Res.*, vol. 28, no. 37, pp. 52295–52305, Oct. 2021.

[3] F. H. Jufri, S. Oh, and J. Jung, "Day-ahead system marginal price forecasting using artificial neural network and similar-days information," *J. Electr. Eng. Technol.*, vol. 14, no. 2, pp. 561–568, Mar. 2019.

[4] J. Lee and Y. Cho, "National-scale electricity peak load forecasting: Traditional, machine learning, or hybrid model?" *Energy*, vol. 239, Jan. 2022, Art. no. 122366.

[5] I. Ilic, B. Görgülü, M. Cevik, and M. G. Baydoğan, "Explainable boosted linear regression for time series forecasting," *Pattern Recognit.*, vol. 120, Dec. 2021, Art. no. 108144.

[6] J. Li, Y. Lei, and S. Yang, "Mid-long term load forecasting model based on support vector machine optimized by improved sparrow search algorithm," *Energy Rep.*, vol. 8, pp. 491–497, Aug. 2022.

[7] F. Wu, C. Cattani, W. Song, and E. Zio, "Fractional ARIMA with an improved cuckoo search optimization for the efficient short-term power load forecasting," *Alexandria Eng. J.*, vol. 59, no. 5, pp. 3111–3118, Oct. 2020.

[8] X. Chen, X. Li, and R. Li, "Ultrasonic power load forecasting based on BP neural network," *J. Inst. Eng. India, Ser. C*, vol. 101, no. 2, pp. 383–390, Apr. 2020.

[9] N. A. Mohammed and A. Al-Bazi, "An adaptive backpropagation algorithm for long-term electricity load forecasting," *Neural Comput. Appl.*, vol. 34, no. 1, pp. 477–491, Jan. 2022.

[10] Z. Chen, Y. Chen, T. Xiao, H. Wang, and P. Hou, "A novel short-term load forecasting framework based on time-series clustering and early classification algorithm," *Energy Buildings*, vol. 251, Nov. 2021, Art. no. 111375.

[11] A. S. Nair, T. Hossen, M. Campion, and P. Ranganathan, "Optimal operation of residential EVs using DNN and clustering based energy forecast," in *Proc. North Amer. Power Symp. (NAPS)*, Sep. 2018, pp. 1–6.

[12] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load forecasting based on LSTM recurrent neural network," *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 841–851, Jan. 2019.

[13] L. Xu, C. Li, X. Xie, and G. Zhang, "Long-short-term memory network based hybrid model for short-term electrical load forecasting," *Information*, vol. 9, no. 7, p. 165, Jul. 2018.

[14] B. Liu, C. Fu, A. Bielefield, and Y. Q. Liu, "Forecasting of Chinese primary energy consumption in 2021 with GRU artificial neural network," *Energies*, vol. 10, no. 10, pp. 1–15, 2017.

[15] X. Tang, Y. Dai, T. Wang, and Y. Chen, "Short-term power load forecasting based on multi-layer bidirectional recurrent neural network," *IET Gener., Transmiss. Distrib.*, vol. 13, no. 17, pp. 3847–3854, Sep. 2019.

[16] M. Zhang, Z. Yu, and Z. Xu, "Short-term load forecasting using recurrent neural networks with input attention mechanism and hidden connection mechanism," *IEEE Access*, vol. 8, pp. 186514–186529, 2020.

[17] A. M. Tudose, D. O. Sidea, I. I. Picioroaga, V. A. Boicea, and C. Bulac, "A CNN based model for short-term load forecasting: A real case study on the Romanian power system," in *Proc. 55th Int. Universities Power Eng. Conf. (UPEC)*, Sep. 2020, pp. 1–6.

[18] S. Luo, Y. Rao, J. Chen, H. Wang, and Z. Wang, "Short-term load forecasting model of distribution transformer based on CNN and LSTM," in *Proc. IEEE Int. Conf. High Voltage Eng. Appl. (ICHVE)*, Sep. 2020, pp. 2020–2023.

[19] C. Ren, L. Jia, and Z. Wang, "A CNN-LSTM hybrid model based short-term power load forecasting," in *Proc. Power Syst. Green Energy Conf. (PSGEC)*, Aug. 2021, pp. 182–186.

[20] M. Sajjad, Z. A. Khan, A. Ullah, T. Hussain, W. Ullah, M. Y. Lee, and S. W. Baik, "A novel CNN-GRU-based hybrid approach for short-term residential load forecasting," *IEEE Access*, vol. 8, pp. 143759–143768, 2020.

[21] R. Zhichao, C. Chao, D. Yingying, Z. Wentao, W. Jun, and Z. Ruixiao, "Short-term load forecasting of multi-layer LSTM neural network considering temperature fuzzification," in *Proc. IEEE Sustain. Power Energy Conf. (iSPEC)*, Nov. 2020, pp. 2398–2404.

[22] H. Bian, Y. Zhong, J. Sun, and F. Shi, "Study on power consumption load forecast based on K-means clustering and FCM–BP model," *Energy Rep.*, vol. 6, pp. 693–700, Dec. 2020.

[23] K. Aurangzeb, M. Alhussein, K. Javaid, and S. I. Haider, "A pyramid-CNN based deep learning model for power load forecasting of similar-profile energy customers based on clustering," *IEEE Access*, vol. 9, pp. 14992–15003, 2021.

[24] X. Dong, L. Qian, and L. Huang, "Short-term load forecasting in smart grid: A combined CNN and K-means clustering approach," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Feb. 2017, pp. 119–125.

[25] Y. Jin, M. A. Acquah, M. Seo, and S. Han, "Short-term electric load prediction using transfer learning with interval estimate adjustment," *Energy Buildings*, vol. 258, Mar. 2022, Art. no. 111846.

[26] T. Yu and H. Zhu, "Hyper-parameter optimization: A review of algorithms and applications," 2020, *arXiv:2003.05689*.

[27] V. Kadappa and A. Negi, "Divide and conquer framework with feature partitioning concepts," in *Proc. IEEE Punecon*, Nov. 2018, pp. 1–5.

[28] M. Weber, M. Turowski, H. K. Cakmak, R. Mikut, U. Kuhnapfel, and V. Hagenmeyer, "Data-driven copy-paste imputation for energy time series," *IEEE Trans. Smart Grid*, vol. 12, no. 6, pp. 5409–5419, Nov. 2021.

[29] D. Gan, Y. Wang, S. Yang, and C. Kang, "Embedding based quantile regression neural network for probabilistic load forecasting," *J. Modern Power Syst. Clean Energy*, vol. 6, no. 2, pp. 244–254, Mar. 2018.

[30] M. J. Alrawashdeh, "An adjusted Grubbs' and generalized extreme studentized deviation," *Demonstratio Mathematica*, vol. 54, no. 1, pp. 548–557, Dec. 2021.

[31] S. Di, "Power system short term load forecasting based on weather factors," in *Proc. 3rd World Conf. Mech. Eng. Intell. Manuf. (WCMEIM)*, Dec. 2020, pp. 694–698.

[32] F. Ziel, "Modeling public holidays in load forecasting: A German case study," *J. Modern Power Syst. Clean Energy*, vol. 6, no. 2, pp. 191–207, Mar. 2018.

[33] C.-H. Wang, G. Grozev, and S. Seo, "Decomposition and statistical analysis for regional electricity demand forecasting," *Energy*, vol. 41, no. 1, pp. 313–325, May 2012.

[34] E. L. Taylor, "Short-term electrical load forecasting for an institutional/industrial power system using an artificial neural network," M.S. thesis, Univ. Tennessee-Knoxville, Knoxville, TN, USA, 2013. [Online]. Available: http://trace.tennessee.edu/utk_gradthes/2468

[35] D. Edelmann, T. F. Móri, and G. J. Székely, "On relationships between the Pearson and the distance correlation coefficients," *Statist. Probab. Lett.*, vol. 169, Feb. 2021, Art. no. 108960.

[36] F. Han, T. Pu, M. Li, and G. Taylor, "Short-Term forecasting of individual residential load based on deep learning and K-means clustering," *CSEE J. Power Energy Syst.*, vol. 7, no. 2, pp. 261–269, 2021.

[37] F. Pourkamali-Anaraki and S. Becker, "Preconditioned data sparsification for big data with applications to PCA and K-means," *IEEE Trans. Inf. Theory*, vol. 63, no. 5, pp. 2954–2974, May 2017.

[38] Y. Li and H. Wu, "A clustering method based on K-means algorithm," *Phys. Proc.*, vol. 25, pp. 1104–1109, Jan. 2012.

[39] C. Johnpaul, M. V. N. K. Prasad, S. Nickolas, and G. R. Gangadharan, "Trendlets: A novel probabilistic representational structures for clustering the time series data," *Expert Syst. With Appl.*, vol. 145, May 2020, doi: 10.1016/j.eswa.2019.113119.

[40] M. Okawa, "Online signature verification using locally weighted dynamic time warping via multiple fusion strategies," *IEEE Access*, vol. 10, pp. 40806–40817, 2022.

[41] T.-Y. Kim and S.-B. Cho, "Predicting residential energy consumption using CNN-LSTM neural networks," *Energy*, vol. 182, pp. 72–81, Sep. 2019.

[42] C. Yu, Y. Li, Q. Chen, X. Lai, and L. Zhao, "Matrix-based wavelet transformation embedded in recurrent neural networks for wind speed prediction," *Appl. Energy*, vol. 324, Oct. 2022, Art. no. 119692.

[43] X. Wang, F. Fang, X. Zhang, Y. Liu, L. Wei, and Y. Shi, "LSTM-based short-term load forecasting for building electricity consumption," in *Proc. IEEE 28th Int. Symp. Ind. Electron. (ISIE)*, Jun. 2019, pp. 1418–1423.

[44] Z. Fazlipour, E. Mashhour, and M. Joorabian, "A deep model for short-term load forecasting applying a stacked autoencoder based on LSTM supported by a multi-stage attention mechanism," *Appl. Energy*, vol. 327, Dec. 2022, Art. no. 120063.

[45] N. B. Bynagari, "The difficulty of learning long-term dependencies with gradient flow in recurrent nets," *Eng. Int.*, vol. 8, no. 2, pp. 127–138, Dec. 2020.

[46] J. Lin, J. Ma, J. Zhu, and Y. Cui, "Short-term load forecasting based on LSTM networks considering attention mechanism," *Int. J. Electr. Power Energy Syst.*, vol. 137, May 2022, Art. no. 107818.

[47] S. Chowdhury and M. P. Schoen, "Research paper classification using supervised machine learning techniques," in *Proc. Intermountain Eng., Technol. Comput. (IETC)*, Oct. 2020, pp. 1–6.

[48] G. Ke, "LightGBM: A highly efficient gradient boosting decision tree," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2017, pp. 3147–3155.

[49] C. S. Bojer and J. P. Meldgaard, "Learnings from Kaggle's forecasting competitions," 2020, *arXiv:2009.07701*.

[50] Korea Power Exchange. (2022). *Public Power Supply and Demand Sharing System*. Accessed: Jan. 15, 2022. [Online]. Available: https://openapi.kpx.or.kr/sukub.do

[51] (2022). *Korea Public Data Portal*. Accessed: Aug. 17, 2021. [Online]. Available: https://www.data.go.kr/en/data/15058629/openapi.do

[52] M. A. Syakur, B. K. Khotimah, E. M. S. Rochman, and B. D. Satoto, "Integration K-Means clustering method and elbow method for identification of the best customer profile cluster," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 336, no. 1, 2018, Art. no. 012017.

[53] F. Batool and C. Hennig, "Clustering with the average silhouette width," *Comput. Statist. Data Anal.*, vol. 158, Jun. 2021, Art. no. 107190.

[54] S. K. Kingrani, M. Levene, and D. Zhang, "Estimating the number of clusters using diversity," *Artif. Intell. Res.*, vol. 7, no. 1, p. 15, Dec. 2017.

[55] M. A. Acquah and S. Han, "Online building load management control with plugged-in electric vehicles considering uncertainties," *Energies*, vol. 12, no. 8, p. 1436, Apr. 2019.

[56] M. K. J. Robertson, *An Introduction to Modern Statistical Methods in HCI*, 1st ed. New York, NY, USA: Springer, 2016.

[57] T. Cerquitelli, G. Malnati, and D. Apiletti, "Exploiting scalable machine-learning distributed frameworks to forecast power consumption of buildings," *Energies*, vol. 12, no. 15, p. 2933, Jul. 2019.

[58] S. J. Taylor and B. Letham, "Forecasting at scale," *Amer. Statistician*, vol. 72, no. 1, pp. 37–45, 2018.

MOSES AMOASI ACQUAH (Member, IEEE) received the B.Sc. and M.Phil. degrees in computer engineering from the University of Ghana, in 2011 and 2014, respectively, and the Ph.D. degree in electrical engineering from Kyungpook National University, South Korea, in 2018. He is currently an Assistant Professor with Keimyung University, Daegu, South Korea. His research interests include energy management systems, vehicle-to-grid systems, big data, the IoT, machine learning, and system optimization.

YUWEI JIN (Member, IEEE) received the M.Sc. degree from the School of Electrical Engineering, Northeast Electric Power University (NEEPU), Jilin, China, in 2017. She is currently pursuing the Ph.D. degree in electrical engineering with Kyungpook National University, Daegu, South Korea. Her research interests include smart grids, V2G, time series forecasting, machine learning, and ESS.

BYEONG-CHAN OH (Graduate Student Member, IEEE) received the B.E. and M.S. degrees from the Department of Electronic and Electrical Engineering, Keimyung University, Daegu, South Korea, in 2018 and 2020, respectively, where he is currently pursuing the Ph.D. degree. His research interests include power system planning, operations of the microgrid, and renewable energy sources.

YEONG-GEON SON (Graduate Student Member, IEEE) received the B.E. and M.S. degrees from the Department of Electronic and Electrical Engineering, Keimyung University, Daegu, South Korea, in 2020, where he is currently pursuing the Ph.D. degree in electronic and electrical engineering. His research interests include power system planning, operations of the microgrid, and power-to-gas systems.

SUNG-YUL KIM (Member, IEEE) received the B.S. and M.Phil. degrees in electrical engineering from Hanyang University, Seoul, South Korea, in 2007 and 2012, respectively. From 2012 to 2013, he was a Research Assistant at the Georgia Institute of Technology, Atlanta, GA, USA. Since 2013, he has been with the Department of Energy Engineering, Keimyung University, Daegu, South Korea. His main research interests include computer aided optimization, renewable energy sources applied to smart grid, and power system reliability.